



*Facultad  
de  
Ciencias*

# **Predicción de lluvia a muy corto plazo mediante el uso de técnicas de aprendizaje profundo**

A DEEP-LEARNING MODEL FOR PRECIPITATION NOWCASTING

*Trabajo de Fin de Máster  
para acceder al*

**Máster en Ciencia de Datos**

Autor: Antonio Pérez Velasco (aperez@predictia.es)

Director/es: José Manuel Gutiérrez Llorente

Daniel San Martín Segura

Septiembre 2020

## Resumen

La predicción de precipitación a muy corto plazo es una herramienta fundamental y con numerosas aplicaciones en el ámbito urbano y rural, teniendo un gran impacto en la agricultura, mantenimiento de aguas residuales, actividades industriales, etc. Actualmente, lo más común es que las aplicaciones operativas hagan uso de observaciones de radar y técnicas basadas en la persistencia lagrangiana (o semi-lagrangiana en el caso de introducir modelización estadística). Sin embargo, está produciéndose un aumento en el estudio del problema mediante el uso de algoritmos de aprendizaje profundo.

En el marco de este trabajo, se consideran un conjunto de radares regionales ofrecidos en abierto por la Agencia Estatal de Meteorología (AEMET) para modelizar, a través del uso de técnicas de aprendizaje profundo, la variabilidad espacio-temporal de la precipitación. Además, se han añadido como variable predictora los datos obtenidos mediante simulaciones numéricas del modelo de mesoescala WRF (Weather Research and forecasting) con el objetivo de estudiar su posible valor añadido.

**Palabras clave:** predicción a corto plazo, precipitación, aprendizaje profundo, redes neuronales, convolución

## Abstract

Precipitation nowcasting is a fundamental tool with numerous applications in urban and rural areas, having a great impact on agriculture, wastewater maintenance, industrial activities, etc. Currently, the most common operational applications are using radar observations and techniques based on Lagrangian persistence (or semi-Lagrangian, in case they include statistical modelization). However, there is an increase in the study of the problem through the use of deep learning algorithms.

In the framework of this work, a set of regional radars offered openly by the State Agency of Meteorology (AEMET) are considered to model -through the use of deep learning techniques- the spatio-temporal variability of precipitation. In addition, data obtained through numerical simulations of the mesoscale model WRF (Weather Research and forecasting) have been added as a predictor variable with the aim of studying its possible added value.

**Key words:** nowcasting, precipitation, deep learning, neural networks, convolution

# Índice general

<b>I</b>	<b>El problema y su contexto</b>	<b>1</b>
<b>1.</b>	<b>Introducción</b>	<b>1</b>
1.1.	Motivación . . . . .	1
1.2.	Estado del arte . . . . .	2
1.3.	Objetivos . . . . .	4
<b>II</b>	<b>Extracción y curación de los datos</b>	<b>6</b>
<b>2.</b>	<b>Origen de los datos</b>	<b>6</b>
2.1.	Observaciones de radar de AEMET . . . . .	6
2.2.	WRF (Weather Research and Forecasting) . . . . .	7
<b>3.</b>	<b>Preprocesamiento</b>	<b>9</b>
3.1.	Observaciones radar: De Imagen a NetCDF . . . . .	9
3.1.1.	Georeferenciación . . . . .	9
3.1.2.	Calibración y eliminación de líneas de administración . . . . .	10
3.1.3.	Extracción del campo de precipitación . . . . .	12
3.1.4.	dBZ a mm/h: Marshall-Palmer fórmula . . . . .	12
3.1.5.	Detección de outliers . . . . .	13
3.1.6.	Unión temporal y relleno de tiempos vacíos . . . . .	13
3.2.	WRF: Acoplamiento espacio-temporal . . . . .	15
3.2.1.	Interpolación temporal (t) . . . . .	15
3.2.2.	Interpolación bilineal: (x, y) a (lon, lat) . . . . .	16
<b>III</b>	<b>Técnicas: estructura, alimentación y análisis</b>	<b>18</b>
<b>4.</b>	<b>Modelización</b>	<b>18</b>
4.1.	Técnicas actuales (benchmarking) . . . . .	18
4.1.1.	Extrapolación . . . . .	18
4.1.2.	S-PROG . . . . .	19
4.1.3.	STEPS . . . . .	20
4.2.	Modelos de aprendizaje profundo . . . . .	21
4.2.1.	Introducción al aprendizaje profundo . . . . .	21
4.2.2.	Métodos y estructuras empleados . . . . .	23
4.2.2.1.	Estructura Encoder-Decoder . . . . .	23
4.2.2.2.	Capa de Convolución . . . . .	23
4.2.2.3.	Capa de reducción por máximos . . . . .	24
4.2.2.4.	Capa de incremento de la dimensionalidad . . . . .	25
4.2.2.5.	Capa de apagado de neuronas . . . . .	25

---

4.2.2.6.	Conexiones entre capas no contiguas . . . . .	26
4.2.3.	Arquitectura de los modelos diseñados . . . . .	26
4.2.4.	Entrenamiento . . . . .	28
4.2.4.1.	Experimento: muestras y set-up . . . . .	28
4.2.4.2.	Función de coste . . . . .	29
4.2.4.3.	Algoritmo de optimización . . . . .	30
4.2.5.	Evaluación . . . . .	31
4.2.5.1.	Predicción iterativa . . . . .	31
4.2.5.2.	Métricas . . . . .	31
<b>5.</b>	<b>Resultados y análisis</b>	<b>33</b>
5.1.	Evolución de los modelos durante la etapa de entrenamiento . . . . .	33
5.2.	Evaluación sobre periodo de tiempo . . . . .	34
5.3.	Evaluación sobre casos de estudio . . . . .	42
5.4.	Efecto del cuantil elegido en la función de coste . . . . .	44
<b>IV</b>	<b>Conclusiones</b>	<b>46</b>
<b>6.</b>	<b>Conclusiones y trabajos futuros</b>	<b>46</b>

## Parte I

# El problema y su contexto

# Capítulo 1

## Introducción

### 1.1. Motivación

De acuerdo a la Sociedad Americana de Meteorología (AMS) [1], la precipitación se define como cualquier producto procedente de la condensación del vapor de agua atmosférico que cae desde las nubes sobre un determinado punto de la superficie terrestre en un período de tiempo. Aunque ésta sea una definición generalista, no se deben pasar por alto los numerosos procesos físicos que toman parte en el complejo proceso de formación de las gotas de lluvia. Cabe destacar que la precipitación es una variable que se puede estudiar tanto de forma continua (unidades en el Sistema Internacional  $\text{kg m}^{-2} \text{s}^{-1}$ , también se usan  $\text{mm h}^{-1}$ ) como de forma discreta (ocurrencia o no de precipitación dado un determinado límite predefinido, del orden de  $0.1 \text{ mm/h}$ ).

Existen muchos sectores donde la precipitación se caracteriza por ser un factor relevante de estudio. No solo porque tenga una influencia en nuestro día a día, sino que da lugar a algunos de los eventos meteorológicos más severos como pueden ser inundaciones, sequías... que además esperan modificar su frecuencia de aparición con el cambio climático [2]. Estos eventos meteorológicos extremos, hasta ahora de baja frecuencia, tienden a ser devastadores en términos socioeconómicos. Debido a ello se han desarrollado planes de contingencia y sistemas de alerta temprana de estos eventos, siendo fundamental una predicción robusta y acertada tanto a muy corto plazo como a largo plazo.

Entre estos sistemas están aquellos dedicados a la predicción de precipitaciones a muy corto plazo, también conocido como *nowcasting* [3]. Este término hace referencia a la predicción durante períodos de tiempo cortos, entre cero y seis horas, de la localización y magnitud de los fenómenos de lluvia a partir del uso de datos de radares, satélites y otras técnicas de teledetección. En la actualidad, existen redes de radares meteorológicos en la mayoría de países europeos, así como en América y parte de Asia. La ventaja principal en el uso de radares es su alta resolución espacio-temporal, siendo del orden de  $1 \text{ km}^2$  con un tiempo de actualización de entre 5 y 10 minutos.

Estas técnicas permiten la caracterización de eventos individuales (como pequeñas tormentas) que en algunos casos no son capturados por las simulaciones numéricas (NWP) convencionales (ver Figura 1.1), aunque estas vivan una revolución paralela en términos de eficiencia, resolución y precisión [4]. La pérdida de capacidad predictiva de la técnicas de predicción a muy corto plazo basadas en el uso de imágenes de radar está directamente relacionada con el carácter caótico de la atmósfera. Las predicciones mediante el uso de observaciones de radar no reflejan los procesos físicos (condiciones de contorno e iniciales, microfísica, orografía ...) que influyen en la evolución de la mancha de precipitación. De modo que surgieron los métodos que combinan los campos de radar con las predicciones de modelos numéricos [5].

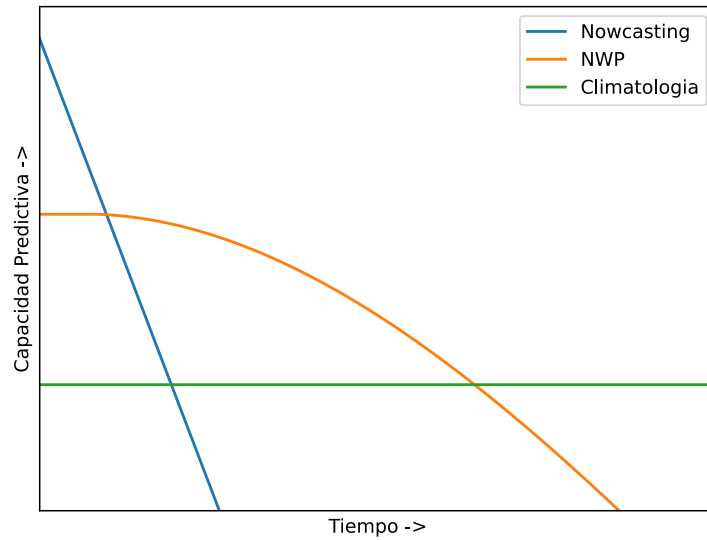


Figura 1.1: Relación entre la capacidad predictiva y el horizonte de predicción de las diferentes técnicas de predicción [6].

Hoy en día existen numerosos sistemas de nowcasting operativos en todo el mundo [7], tanto a nivel regional como nacional. La mayoría de ellos se apoyan en la persistencia lagrangiana considerando que la atmósfera cambia de la misma manera que lo hacía en el pasado cercano para extrapolar el campo de precipitación de las observaciones más recientes al futuro, o en un esquema semi-lagrangiano basado en el uso de técnicas estadísticas que parten de la hipótesis mencionada.

Además, se pueden distinguir dos enfoques: determinista y probabilista, donde la segunda corriente nace a raíz de la necesidad de hacer medible los diferentes errores de medición que procedan del radar meteorológico (falsos ecos, calibración defectuosa y error en la transformación de unidades, entre otros [8]).

A pesar del creciente uso de algoritmos de aprendizaje automático y aprendizaje profundo en numerosos sectores del ámbito científico, no ha sido hasta hace poco cuando se ha producido una explosión sobre el uso de dichas técnicas en los problemas de nowcasting de precipitación que reunían todas las condiciones necesarias para ser altamente modelizables.

Por ello, se decidió hacer uso de las observaciones proporcionadas a través de la red de radares regionales de la Agencia Estatal de Meteorología (AEMET) para proponer la creación de un modelo que permita capturar el movimiento y magnitud de la mancha de precipitación tratando la variable de forma continua.

## 1.2. Estado del arte

Como se ha explicado anteriormente, existen numerosas redes abiertas de radares regionales y/o nacionales a lo largo del mundo. Prueba de ello es la aplicación comercial [Rainviewer](#), que hace uso de los datos en abierto comentados anteriormente para ofrecer la precipitación in-vivo de las zonas alcanzadas por los radares de los que dispone de acceso.

El uso de observaciones de radar requiere un proceso de corrección y limpieza debido a los errores de medición del radar meteorológico. Normalmente, la receta de corrección de las observaciones originales suele incluir varias tareas:

- Eliminación de falsos ecos: los radares no solo son capaces de recibir información acerca del estado de la lluvia, sino que al funcionar por efecto Doppler también son capaces de detectar otros objetos en movimiento como pueden ser bandadas de pájaros. [Enlazamos](#) una curiosa noticia acerca de la detección de un enjambre de hormigas voladoras con observaciones de radar.
- Corrección de la atenuación: especialmente en radares que se mueven en las bandas C (longitud de onda de 5cm (4-8 GHz)) y X (caso de la red regional de radares de AEMET [9], con una longitud de onda de 3 cm (8-12 GHz)). Esta atenuación de la mancha de precipitación produce que exista una subestimación del valor real de precipitación y por tanto el valor de la observación se pueda mover un rango de valores de menor magnitud que el valor real.
- Transformación de unidades: generalmente de dBZ a mm/h, se hablará más adelante de la aproximación más adecuada para realizar dicha transformación.

Existen librerías de software capaces de llevar a cabo dicho proceso, como [Wradlib](#) [10]. En este trabajo no se ha considerado dicho preprocesado de las observaciones de radar meteorológico debido a que AEMET ya incluyó un procesador de datos de nueva generación que permitía un aumento de la calidad de productos finales tales como el filtrado de ecos no meteorológicos y la estimación de la precipitación entre 2006 y 2011 [9].

Cabe destacar que [Rainviewer](#) no solo muestra la precipitación en tiempo real y los resultados de un modelo numérico para las próximas 48 horas, sino que además ha añadido un modulo de predicción a corto plazo (2 horas) usando técnicas basadas en la persistencia lagrangiana.

Estas son las más comunes y la mayoría de aplicaciones disponibles hoy en día están implementadas mediante el uso de estas técnicas estadísticas, aunque como se hablará más adelante, actualmente existen numerosas investigaciones en el campo del aprendizaje automático y profundo. Es por ello, que existen módulos y paquetes diversos en diferentes lenguajes de programación que contienen algoritmos, tanto deterministas como probabilistas, basados en la persistencia lagrangiana. Como ejemplo, se mencionan [Rainymotion](#) [11] y [Pysteps](#) [12], ambos en Python.

Aunque las técnicas estadísticas siguen en un proceso continuo de mejora e investigación, tanto en la mejora de algoritmos como en la inclusión de nuevas variables predictoras [13], no son capaces de aprovechar la gran cantidad de imágenes de las que se disponen hoy en día porque se basan en ecuaciones de parámetros constantes. Sin embargo, el problema de nowcasting reúne todos los ingredientes necesarios para ser modelizable mediante el uso de técnicas de aprendizaje profundo [14] debido a su alta dimensionalidad en las componentes espacial y temporal, además del creciente uso de unidades de procesamiento más eficientes (GPUs y TPUs).

Al evaluar la bibliografía, se ha percibido un aumento en el uso de técnicas de aprendizaje profundo en las investigaciones más recientes acerca del problema de estudio. Teniendo como objetivo el captar la variabilidad espacio-temporal de la precipitación, de acuerdo con la literatura lo más indicado sería capturar a-priori la componente espacial con redes convolucionales CNN (del inglés *Convolutional Neural Networks*) y estudiar la componente temporal con una red que memorice relaciones en largos y cortos periodos de tiempo, como son las LSTM [15] (del inglés *Long-Short Term Memory*).

Por ello, uno de los primeros acercamientos conllevó a la proposición de un nuevo tipo de estructura llamada ConvLSTM [16], que capturaba mejor las relaciones espacio-temporales que la propia LSTM y proporcionaba una predicciones más precisas que las obtenidas mediante técnicas de persistencia lagrangiana. A pesar de proponer una estructura novedosa para la resolución de problemas con componentes espacial y temporal, la manera de realizar el entrenamiento no fue la más adecuada ya que utilizaban únicamente 97 muestras de días lluviosos y un único límite para la definición de ocurrencia de precipitación igual a 0.5 mm/h con lo que no se estudió la capacidad de la red a observar eventos más precipitantes como son las tormentas. Unos años más tarde, el mismo grupo



de investigación llevó el problema por un cauce diferente utilizando un modelo que añadiera un filtro que considerara como falsa la invarianza en la localización que habían considerado con las ConvLSTM, llamándolo TrajGRU (del inglés *Trajectory Gate Recurrent Unit*). En otras palabras: no solo se tiene en cuenta la posición actual de la precipitación si no que se tienen en cuenta la localización en los puntos temporales anteriores [17]. Otros autores también han utilizado este mismo método [18].

Otros autores -además de hacer énfasis en la estructura de red utilizada- han estudiado las diferencia de resultados al usar diferentes tipos de escalado, funciones de coste y optimizadores [19], de especial relevancia en un problema de este tipo como se comentará más adelante. Aunque se haya dicho que sería recomendable el uso de LSTM para afrontar el problema de la componente temporal, este mismo autor ha desarrollado un modelo basado únicamente en convoluciones llamado *Rainnet* [20], basado en el uso de estructuras tipo U-Net [21] y SegNet [22], cuyos resultados son de gran relevancia en términos de los primeros 60 minutos, superando a los métodos de persistencia lagrangiana y a la persistencia euleriana.

Las grandes empresas han entrado también en el juego. Por las diversas entradas en su [blog](#) dedicado a la inteligencia artificial, es conocido que Google LLC. -una de las tres mayores empresas tecnológicas del momento- está introduciéndose en este tema de manera sobresaliente debido, entre otras cosas, a su alta capacidad de cómputo. La [primera entrada](#) [23] que escribieron sobre este tema presentaba un contenido muy similar al de *Rainnet*, y no fue hasta la [segunda entrada](#) [24] cuando presentaron *MetNet*. Un modelo extraordinariamente complejo con más de 200 millones (frente a los algo más de 30 millones de *Rainnet*) de parámetros que sería capaz de batir durante las primeras 8 horas al modelo numérico utilizado por la NOAA en Estados Unidos. Basándose en la concatenación de las observaciones de radar con los datos proporcionados por el satélite GOES-16, además de otras variables como latitud, longitud, orografía, día, mes y hora para cada tiempo  $t$ ,  $t-15$  minutos,  $t-30$  minutos, ... Antes de pasar los datos a la estructura de red, se realiza una transformación multicategoría configurando las precipitación de variable continua a variable 512 categórica. La estructura de red se basa en una capa que reduce la dimensionalidad en el espacio mediante el uso de convoluciones, una capa ConvLSTM que se encarga de codificar los tiempos y un agregador espacial. Los resultados no tienen precedentes, aunque el gasto computacional es excesivo ya que hacen uso de 256 Google TPUs cuyo coste económico por contrato anual se sitúa en [1.059.610 \\$](#). El objetivo de utilizar dicho número de procesadores tensoriales es la paralelización masiva del proceso de entrenamiento y predicción, lo que hace capaz a *MetNet* de predecir las próximas 8 horas (con una resolución temporal de 2 minutos) para todo Estados Unidos en cuestión de segundos. Obviamente, para este trabajo no se ha hecho uso de una capacidad de cómputo como la de *Metnet*.

Es de interés comentar que este trabajo ha sido realizado en el marco de una empresa cuya actividad se encuentra enfocada principalmente en el ámbito de la predicción meteorológica, motivando la propuesta planteada como una posible fuente de predicción a muy corto plazo.

### 1.3. Objetivos

Además de gestionar un software que permita la descarga y preprocesado de las imágenes de radares ofrecidas por la Agencia Estatal de Meteorología (AEMET), el propósito principal de este trabajo se basa en el uso de técnicas de aprendizaje profundo para desarrollar un modelo que sea capaz de caracterizar la variabilidad espacio-temporal de la precipitación (ver Figura 1.2).

Aunque se ha realizado la descarga de datos para todos los radares regionales disponibles en la red de radares de AEMET, solo algunos con características especiales han sido utilizados. Entre estas características se han tenido en cuenta que estén situados en zonas de especial carácter lluvioso, alejadas de ciudades y montañas que puedan producir falsos ecos y que contengan episodios de

precipitación diversos.

No solo se utilizan los datos de las observaciones de radares meteorológicos mencionados anteriormente, sino que también se ha planteado el uso del modelo de mesoescala WRF (Weather Research and Forecasting) como una variable más.

Se realizará una evaluación de los modelos tanto para casos de estudio concretos como para un periodo completo de precipitación, y constará tanto de métricas que midan la capacidad de predicción usando la precipitación de forma continua (error absoluto medio (MAE), la raíz del error cuadrático medio (RMSE) o la correlación de pearson) como de forma discreta (índice crítico de éxito (CSI)).

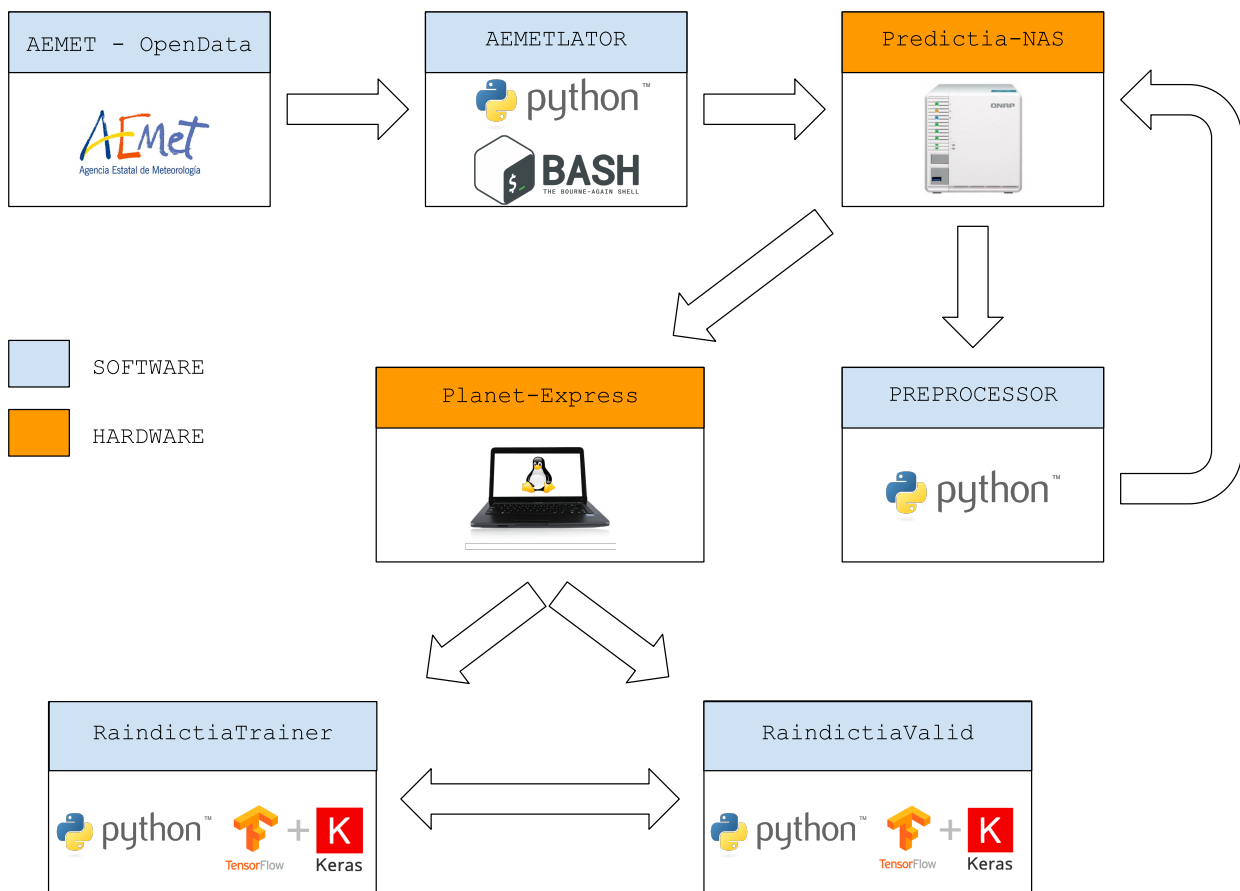


Figura 1.2: Diseño del framework utilizado para la resolución del problema de nowcasting. Descarga, preprocesado, entrenamiento y validación.

## Parte II

# Extracción y curación de los datos

## Capítulo 2

# Origen de los datos

### 2.1. Observaciones de radar de AEMET

La Agencia Estatal de Meteorología (AEMET) ha desarrollado un API REST (*Application Programming Interface. REpresentational State Transfer*) llamada [OpenData](#) con la que se tiene acceso a través de una clave de acceso unipersonal a la descarga gratuita de productos meteorológicos y climáticos.

Entre dichos productos se encuentran los datos de radares regionales, con estaciones en A Coruña, Almería, Asturias, Barcelona, Cáceres, Islas Baleares, Las Palmas, Madrid, Murcia, Málaga, Palencia, Sevilla, Valencia, Vizcaya y Zaragoza, a partir de los cuales AEMET es capaz de obtener el [mapa nacional](#) mostrado en su página web oficial.

El software desarrollado es capaz de descargar en formato GIF los datos de los diferentes radares regionales con un tamaño de imagen que varía en torno a los 25 KiloBytes. Estas imágenes han sido guardadas en un dispositivo de almacenamiento conectado en red (NAS) con una capacidad total de 16 TeraBytes. Teniendo en cuenta que el proceso de descarga se comenzó el 15 de Octubre de 2019 a las 08:00 UTC y que no se ha detenido desde entonces, se han almacenado en total más de 660.000 imágenes ocupando alrededor de 8 GigaBytes. Además del proceso de descarga, el software contiene un sistema de alertas en caso de que dicho proceso no se haya realizado con éxito (ver Figura 2.1). Este sistema se ha desarrollado con el objetivo de conocer caídas del sistema de datos en abierto de AEMET, pero sobre todo para conocer fallos por nuestra parte y poner solución lo antes posible con el objetivo de perder el menor número de observaciones de radar posible.

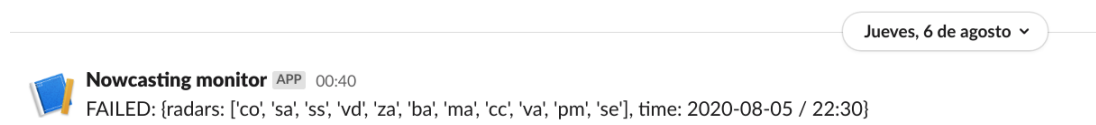


Figura 2.1: Mensaje de Slack (sistema de comunicación empresarial de Predictia Intelligent Data Solutions S.L) avisando de la descarga fallida de varios radares.

Los datos han sido guardados siguiendo la siguiente estructura de ficheros:

```
*/[nombre_radar]/[año]/[mes]/[dia]/nombre_del_fichero.gif
```

donde el nombre del fichero viene dado como:

```
[nombre_radar]_[año][mes][dia][hora][minuto].gif
```

Por ejemplo, el archivo para la observación de radar de Palencia el 22 de Noviembre de 2019 a las 06:20 UTC se nombró como:

\*/vd/2019/11/22/vd\_201911220620.gif

La imagen original tiene un tamaño de 530 x 480 píxeles, que se corresponden con 480 x 480 píxeles de la mancha de precipitación en bruto y 50 x 480 píxeles reservados para la leyenda (ver Figura 2.2). En la leyenda aparecen la banda de colores donde se asocia cada color con un valor de la reflectividad, por tanto la intensidad de precipitación está discretizada en este caso.

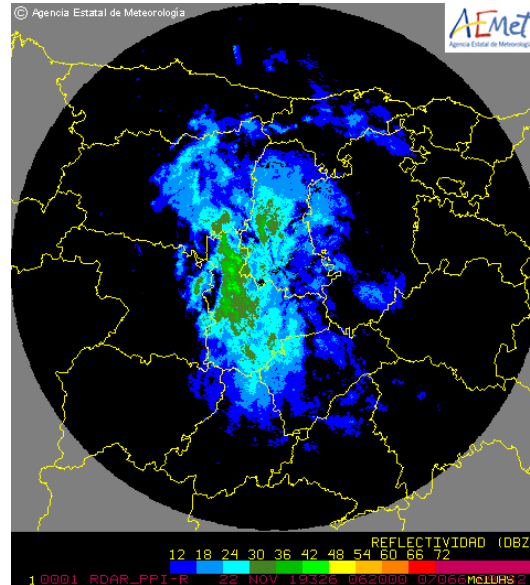


Figura 2.2: Observación de radar directamente descargada a través de la API Rest oficial proporcionada por la Agencia Estatal de Meteorología (AEMET) para su radar localizado en Palencia a fecha 22 de Noviembre de 2019 a las 06:20 UTC.

## 2.2. WRF (Weather Research and Forecasting)

El modelo *Weather Research and Forecasting (WRF)* es un sistema de modelización numérica de la atmósfera desarrollado por el Centro Nacional para la investigación atmosférica (NCAR). Sus aplicaciones pueden utilizar resoluciones espaciales variando desde los metros hasta los miles de kilómetros. Normalmente, se denomina modelo de mesoescala siguiendo las referencias aportadas por la Sociedad Americana de Meteorología [25]. Este tipo de modelos permiten, entre otras cosas, captar dentro del marco de una región determinados fenómenos físicos que en ocasiones los modelos globales pasan por desapercibido. Aunque hoy en día, los modelos globales son cada vez más resolutivos en el eje horizontal (llegando a alcanzar tamaños menores a 5 km, como el caso del modelo global del MetOffice), es conocida la capacidad de los modelos regionales para obtener información sobre tormentas, frentes de aire caliente y frío, bandas de precipitación...

En Predictia Intelligent Data Solutions se hace uso de la aplicación WRF for GRID (WRF4G) [26] diseñada con el objetivo de facilitar la utilización del modelo WRF, incluyendo un sistema de monitorización. Estos modelos son parametrizables, es decir, tienen parámetros configurables para la optimización del modelo según la región o localización sobre la cual se quieran obtener los resultados. En el caso de las simulaciones realizadas por nuestro equipo, presentan dos físicas o parametrizaciones diferentes (operA y operB). Aunque no es de gran relevancia porque para el planteamiento del problema se ha realizado el promedio de las simulaciones obtenidas por ambas físicas, puede ser interesante comentar que las diferencias principales entre ambas parametrizaciones se basan -a grandes rasgos- en el uso de una capa límite planetaria diferente y una microfísica más compleja en el caso de operB [27]. El resto de parámetros permanecen constantes. Para que el experimento sea reproducible, se añaden las físicas al completo utilizadas para dichas simulaciones:



## Capítulo 3

# Preprocesamiento

### 3.1. Observaciones radar: De Imagen a NetCDF

El dato en bruto obtenido del OpenData de AEMET (ver Figura 2.2) no es válido para alimentar a los diferentes modelos utilizados. Por ello, se ha creado un modulo en Python dedicado a la transformación de los datos desde el GIF descargado de AEMET hasta un [netCDF](#) que almacena los datos espacio-temporalmente.

#### 3.1.1. Georeferenciación

En este tipo de problemas es común la georeferenciación de los datos de entrada. La georeferenciación [29] se trata de la asignación pixel a pixel de una coordenada geográfica (latitud y longitud, en el caso de sistemas de coordenadas rectangulares) dentro del marco de un sistema de referencia dado por el metadato de AEMET, en este caso [EPSG: 3857](#). Dicho sistema permite la visualización, consulta y análisis de los datos entre diferentes fuentes de información, siempre y cuando ambas estén referenciadas.

En el caso de la observación del radar situado en Palencia, se tienen las coordenadas geográficas de los cuatro puntos que conforman las esquinas de la imagen sin tener en cuenta la leyenda. En otras palabras, sabiendo que la imagen tiene una resolución de 530 x 480 píxeles, se tienen los valores de los píxeles (0, 0), (0, 480), (480, 0) y (480, 480) que son (-7.61°, 44.23°), (-7.61°, 39.77°), (-1.56°, 44.23°) y (-1.56°, 39.77°), respectivamente. Con dichas coordenadas geográficas y el sistema de referencia utilizado, se asignan a cada uno del resto de píxeles (i, j) una coordenada determinada (ver Figura 3.1). Este proceso se realiza de igual forma para cada uno de los radares de la red de radares regionales de AEMET.

Los módulos y paquetes utilizados para dicha georeferenciación son: *GDAL*, *rasterio* y *xarray*. Al acabar este paso, se obtiene un objeto de *xarray* que almacena las coordenadas espaciales y temporales de la imagen procesada. Este objeto se guarda posteriormente como un fichero *netCDF*. Es decir, cada imagen procesada tendrá un archivo *netCDF* asociado a ella con la siguiente estructura una vez cargado en *xarray*:

```
Dimensions:  (time: 1, x: 480, y: 530)
Coordinates:
  * y        (y) float64  44.23  44.22  ...  39.32  39.31
  * x        (x) float64  -7.609 -7.596 ... -1.591 -1.578
  * time     (time) datetime64[ns] 2019-11-22T00:10:00
Data variables:
  pr        (time, y, x) float64  2.0  2.0  ...  0.0  0.0
```

Así, la variable  $pr$  asociada a la precipitación es dependiente de cada una de las coordenadas espacio-temporales. También, es de notar que la resolución espacial es diferente para cada una de las componentes horizontales ( $x$  e  $y$ ).

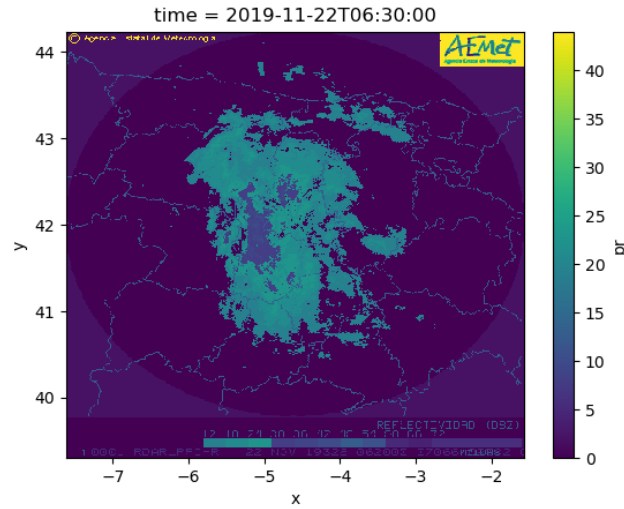


Figura 3.1: Imagen de AEMET para el radar regional de Palencia georeferenciada y abierta con los módulos de Python: Rasterio y Xarray.

### 3.1.2. Calibración y eliminación de líneas de administración

Como se ve en la Figura 3.1, la escala de color no se corresponde con los valores que aparecen en la imagen original de AEMET. Esto se debe a que xarray ha asignado un valor al color independiente de la escala de color que aparece en los 50 x 480 pixeles inferiores. Comparando los valores que aparecen en la escala de color original y los valores asignados por el módulo de Python, se ha hecho una transformación manual de los valores obteniendo lo que se ve en la Figura 3.2.

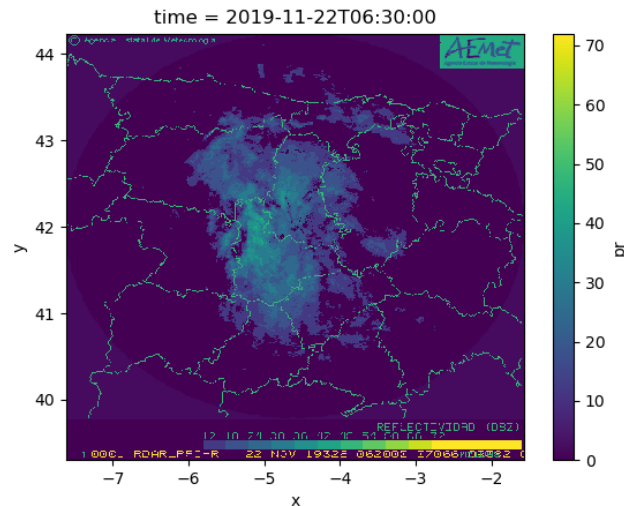


Figura 3.2: Imagen de AEMET después de realizar una calibración color-valor para el radar regional de Palencia georeferenciada, y visualizada con los módulos de Python: Rasterio y Xarray.

No se ha dicho anteriormente, pero en la observación descargada de AEMET existe una coincidencia entre las líneas administrativas, que indican la separación de territorios, y uno de los valores de



reflectividad - precipitación dado en la escala de color, produciendo que sea necesaria la eliminación de las líneas administrativas. Este valor es el 48 dBZ (amarillo). Para realizar esto, se ha escogido una observación en la que no se apreciara mancha de precipitación o que lo poco que se apreciara fuera un falso eco que no interfiera con las líneas de administración (ver Figura 3.3). Una vez escogida, se han eliminado de la Figura 3.2 aquellos puntos cuyo valor se corresponde con una línea de administración en la Figura 3.3.

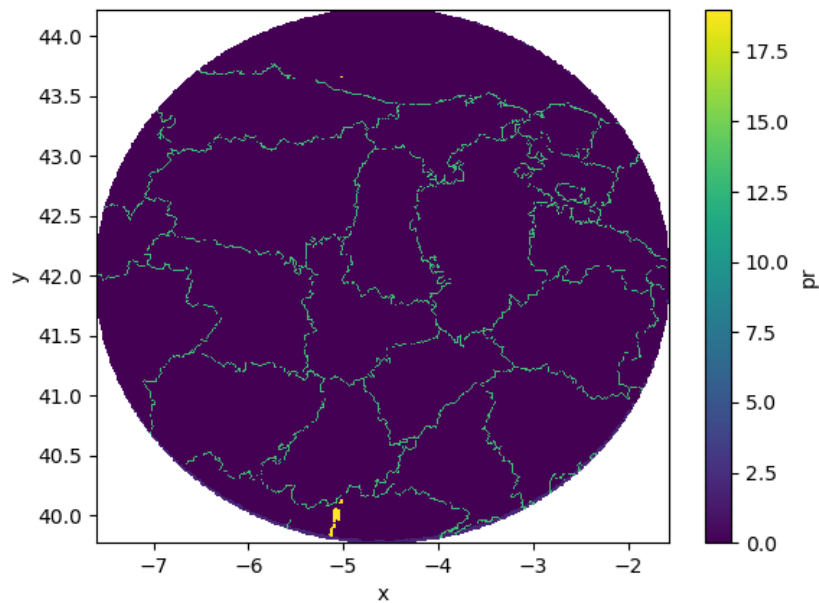


Figura 3.3: Imagen de calibración para el radar regional de Palencia georeferenciada, visualizada con los módulos de Python: Rasterio y Xarray.

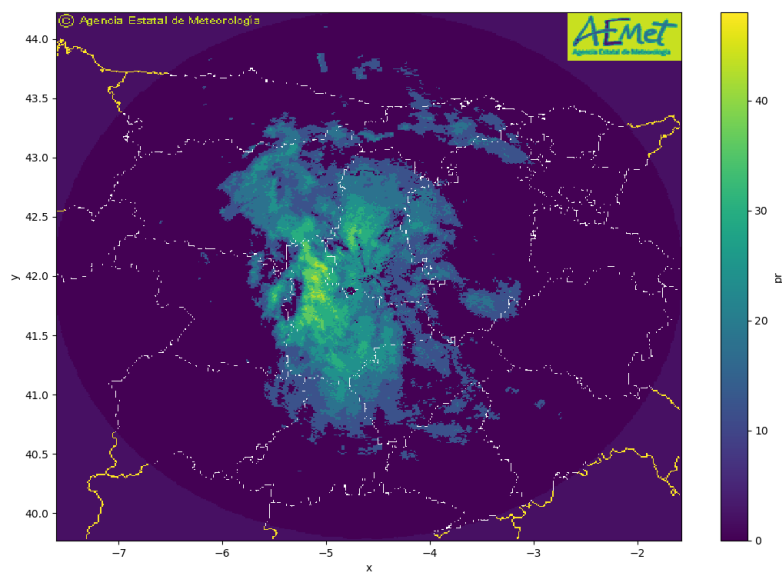


Figura 3.4: Imagen sin líneas administrativas para el radar regional de Palencia, visualizada con los módulos de Python: Rasterio y Xarray.

Una vez eliminadas, se rellenan las líneas de administración con una técnica de vecinos cercanos, es decir, para cada pixel vacío se buscan los  $K=6$  vecinos más cercanos (elegido en base a inspección manual de los valores) y se rellena el valor con el más representativo (ver Figura 3.5).

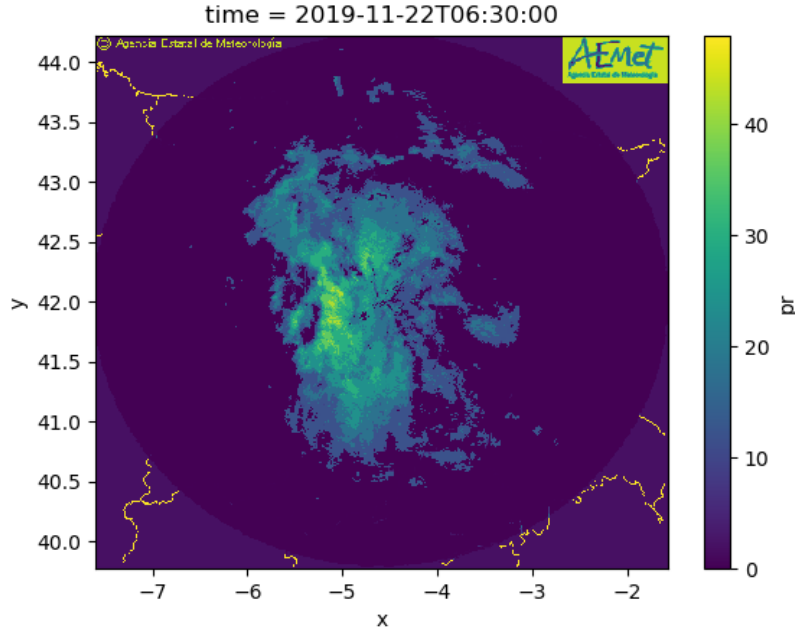


Figura 3.5: Imagen del radar regional de Palencia después de rellenar las líneas administrativas con la técnica de vecinos cercanos, visualizada con los módulos de Python: Rasterio y Xarray.

### 3.1.3. Extracción del campo de precipitación

La Figura 3.5 muestra la mancha de precipitación procesada en el centro de la observación de radar, quedando por eliminar algunos detalles como las líneas de administración y los logotipos de AEMET que se encuentran en el exterior de dicha mancha de precipitación. Esta superficie de precipitación se corresponde con la elipse de color más oscuro que tiene al dispositivo de radar en su centro.

Nos deshacemos de todos los puntos que se encuentren fuera de los límites de la elipse con centro en el dispositivo radar, y añadimos un valor de 0 en dichos puntos.

### 3.1.4. dBZ a mm/h: Marshall-Palmer fórmula

Se ha podido ver en la leyenda de la imagen original de AEMET que las unidades de medida de la cantidad de precipitación son los dBZ (reflectividad). Esta es una unidad técnica, utilizada especialmente en radares meteorológicos, que permite la estimación de la intensidad de precipitación o nieve, siendo proporcional al conteo del número de gotas por unidad de volumen y la sexta potencia del diámetro de dichas gotas. En caso de necesitar la variable de forma continua, se hace uso de la relación de Marshall-Palmer (ver Ecuación 3.1).

$$Ratio = \left( \frac{10^{(dBZ/10)}}{200} \right)^{0.625} \quad (3.1)$$

Sin embargo, como la Ecuación 3.1 utiliza constantes experimentales detalladas para la estimación de lluvia, se llevan a cabo sobre estimaciones debidas al granizo o la banda brillante. En la Tabla 3.1 se muestra la transformación de los valores que aparecen en la leyenda original de AEMET, y

se comprueba que la Ecuación 3.1 realiza sobre estimaciones del ratio de precipitación en el caso de granizo y nieve ( $\text{dBZ} \geq 54$ ).

L (dBZ)	R (mm/h)	R (pulgadas/h)
12	0.205	0.008
18	0.486	0.019
24	1.153	0.045
30	2.734	0.107
36	6.484	0.255
42	15.37	0.605
48	36.46	1.435
54	86.46	3.403
60	205.1	8.074
66	486.2	19.14
72	1153.1	45.39

Tabla 3.1: Conversión de la intensidad de precipitación en dBZ a ratios de caída de la lluvia en mm/h y pulgadas/h mediante la relación de Marshall-Palmer (ver Ecuación 3.1).

### 3.1.5. Detección de outliers

De acuerdo a expertos meteorológicos de Predictia Intelligent Data Solutions, cualquier valor de precipitación superior a 30 mm/h puede ser considerado como un outlier dentro de las características históricas del territorio nacional. Es por esto que se ha añadido esta funcionalidad de detección de puntos en el interior del campo de precipitación con valores superiores a 30 mm/h. Una vez localizados dichos valores, se han eliminado y redefinido a partir de la técnica de vecinos cercanos ya mencionada en la sección de *Calibración y eliminación de líneas de administración*.

### 3.1.6. Unión temporal y relleno de tiempos vacíos

Una vez terminado el procesamiento de una observación de radar (ver Figura 3.6) determinada se almacenan los datos en un fichero netCDF con la misma estructura de ficheros y nombre de archivo que el GIF que se descarga de la API oficial de AEMET.

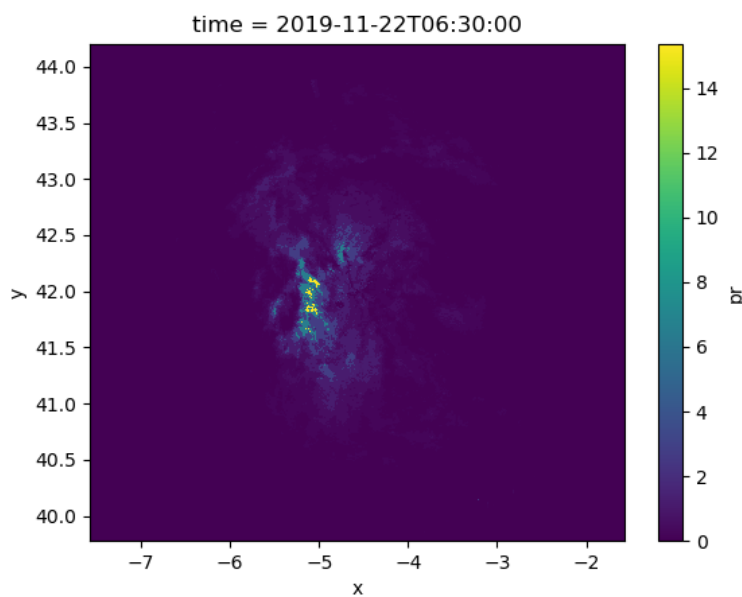


Figura 3.6: Imagen procesada completamente para el radar regional de Palencia, visualizada con el módulo de Python: Xarray.

La decisión de formar archivos espacio-temporales de 15 días implica:

- Crear un fichero netCDF para cada una de las observaciones de radar (frecuencia temporal de 10 minutos) dentro de dicho periodo de 15 días y juntar dichos archivos en un único objeto de xarray
- Comprobar si existen espacios temporales vacíos. Es posible que- bien sea por causa de una caída de conexión de la Máquina Virtual (VM) utilizada para la descarga de datos, por una caída del sistema de gestión de datos de AEMET, por problemas con el dispositivo de medida...- no se haya podido realizar la descarga de ciertas observaciones.
- Rellenar los espacios temporales mediante el uso de técnicas estadísticas basadas en la persistencia lagrangiana (s-prog).
- Guardar el objeto que contiene los datos en forma de fichero netCDF con el nombre siguiendo la estructura:

```
*/[nombre_radar]/[nombre_radar]_[fecha_inicial]_[fecha_final].nc
```

```
dimensions:
```

```
    time = 2304 ;
    x = 477 ;
    y = 477 ;
```

```
variables:
```

```
    double time(time) ;
        time:__FillValue = NaN ;
        time:long_name = "time" ;
        time:standard_name = "time" ;
        time:units = "days_since_2019-01-01" ;
        time:calendar = "proleptic_gregorian" ;
    double x(x) ;
        x:__FillValue = NaN ;
        x:units = "degrees_east" ;
        x:standard_name = "longitude" ;
        x:long_name = "longitude" ;
    double y(y) ;
        y:__FillValue = NaN ;
        y:units = "degrees_north" ;
        y:standard_name = "latitude" ;
        y:long_name = "latitude" ;
    double pr(time, y, x) ;
        pr:__FillValue = NaN ;
        pr:units = "kg_m-2_h-1" ;
        pr:long_name = "Precipitation" ;
        pr:standard_name = "precipitation_flux" ;
```

```
// global attributes:
```

```
    :Conventions = "CF-1.4" ;
    :contact = "aperez@predictia.es" ;
```

Una vez producido dicho conjunto de datos espacio-temporales se intenta reducir la dimensionalidad espacial sin perder gran representatividad eligiendo los puntos de la dimensión x que se encuentran entre el píxel [38,438], e igualmente para la dimensión y. Esto se realiza porque la mayoría de puntos

fuera de dichos rangos tienen valor 0. El tiempo total de procesado para un dataset de 15 días llega a ser de 30 minutos.

Este procesamiento ha sido realizado con todos los radares de la red de radares regionales de AEMET. Aunque muchos de ellos han sido descartados (como se comentará más adelante) para la modelización de datos debido a, entre otras cosas, inconsistencia de los datos, falta de datos, repetición de imágenes por mal estado del radar, observaciones con gran cantidad de ruido y falsos ecos y falta de tiempo de entrenamiento.

## 3.2. WRF: Acoplamiento espacio-temporal

Como se ha explicado en el Capítulo de *Origen de los datos*, los datos generados a partir de las simulaciones numéricas de WRF (Weather Research and Forecasting) tienen una resolución espacial y temporal de 8 km y 1 hora, respectivamente. Por tanto, se hace necesario el aumento de la resolución de los datos de WRF para poder ser acoplados a las observaciones de radar.

### 3.2.1. Interpolación temporal (t)

En primer lugar, se ha realizado una interpolación de los datos horarios mediante el uso del paquete de Python llamado *PySteps*. Este paquete nos permite utilizar técnicas estadísticas basadas en la persistencia lagrangiana para pasar de una resolución temporal de 1h ( $t$ ,  $t + 60\text{min}$ ,  $t + 120\text{min}$ ,...) a la resolución temporal de las observaciones de radar, es decir, 10 minutos ( $t$ ,  $t + 10\text{min}$ , ...,  $t + 60\text{min}$ ,...) (ver Figura 3.7).

Calculando el movimiento del campo de precipitación en instantes anteriores y utilizando la localización de la mancha de precipitación en el momento  $t + 60\text{min}$ , es posible obtener los valores temporales intermedios con frecuencia diezminutal.

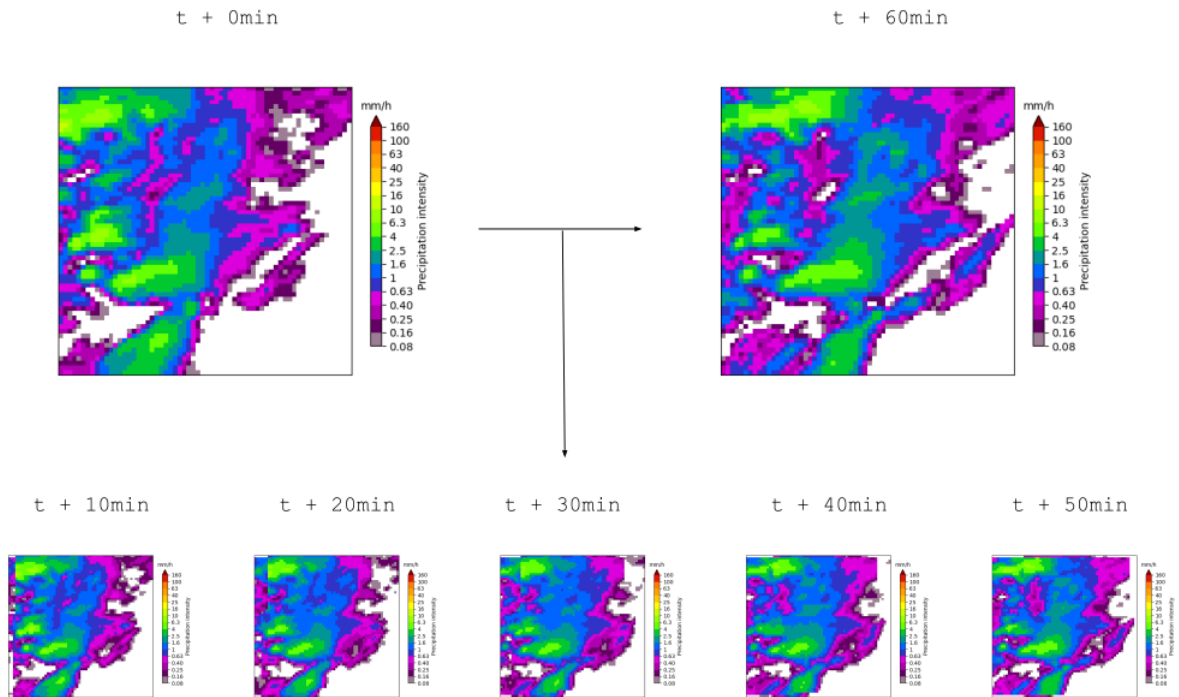


Figura 3.7: Esquema de la interpolación lineal realizada sobre los datos del modelo numérico WRF mediante el uso de PySteps. Arriba se muestran los datos para la simulación numérica dentro del dominio del radar de Palencia del 22-11-2019 06:00 y del 22-11-2019 07:00, mientras que abajo se muestran los datos generados por interpolación.

### 3.2.2. Interpolación bilineal: (x, y) a (lon, lat)

El componente espacial de la precipitación se puede visualizar como una malla de puntos donde a cada píxel se le asigna un valor determinado. Se llama tamaño de malla a la distancia entre el centro de dos píxeles.

Dado un problema en el que se tienen datos con tamaño de malla diferentes se requiere del uso de un algoritmo de interpolación para otorgar a los datos con una menor resolución (mayor tamaño de malla) la misma resolución que los otros datos con los que se quiere trabajar, comúnmente llamado *regridding* [30]. En el caso del problema que estamos afrontando, las observaciones de radar presentan una resolución espacial cercana a 1 km en una malla rectilínea mientras que los datos de WRF son procesados con una resolución espacial de en torno a 8 km en una malla curvilínea (ver Figura 3.8).

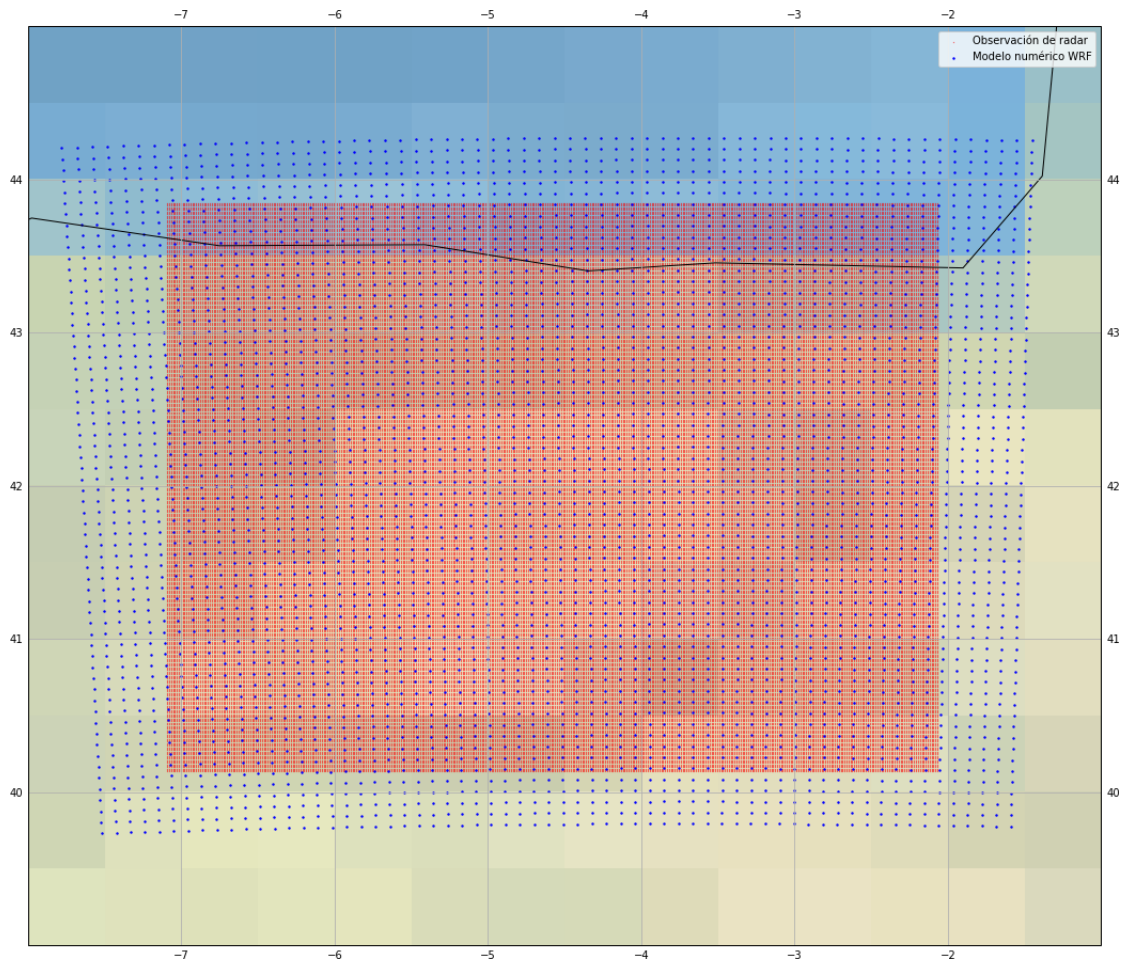


Figura 3.8: Esquema de puntos para el dominio del radar de Palencia en el caso de las observaciones de radar (rojo) y las simulaciones numéricas de WRF (azul).

Existen numerosos métodos de interpolación de mallas y se debe elegir el método apropiado para la tarea prevista. El uso de un esquema de interpolación inapropiado puede conducir a resultados erróneos.

Entre estos métodos se encuentran: interpolación lineal, interpolación bilineal, método conservativo, interpolación a vecinos cercanos, ...

El método de interpolación bilineal ha sido elegido para esta tarea, con el objetivo de convertir los datos curvilíneos de WRF a una malla rectilínea definida con la misma resolución que las observaciones de radar. Esta metodología es una extensión de la interpolación lineal ampliamente

conocida, cuyo objetivo se basa en la caracterización a través de una función  $f(x, y)$  del dominio definido en  $x_0 < x < x_1$  y  $y_0 < y < y_1$ , donde los valores de  $x_0$ ,  $y_0$ ,  $x_1$  e  $y_1$  vienen dados por las muestras conocidas. Los pasos son los siguientes:

- Realizar una interpolación lineal en la dimensión  $x$  (longitudes):

$$f(x, y_0) \approx \frac{x_1 - x}{x_1 - x_0} f(x_0, y_0) + \frac{x - x_0}{x_1 - x_0} f(x_1, y_0)$$

$$f(x, y_1) \approx \frac{x_1 - x}{x_1 - x_0} f(x_0, y_1) + \frac{x - x_0}{x_1 - x_0} f(x_1, y_1)$$

- Utilizar la aproximación obtenida para interpolar en la dimensión  $y$  (latitudes):

$$\begin{aligned} f(x, y) &\approx \frac{y_1 - y}{y_1 - y_0} f(x, y_0) + \frac{y - y_0}{y_1 - y_0} f(x, y_1) \\ &= \frac{1}{(x_1 - x_0)(y_1 - y_0)} \begin{bmatrix} x_1 - x & x - x_0 \end{bmatrix} \begin{bmatrix} f(x_0, y_0) & f(x_0, y_1) \\ f(x_1, y_0) & f(x_1, y_1) \end{bmatrix} \begin{bmatrix} y_1 - y \\ y - y_0 \end{bmatrix} \end{aligned}$$

En el caso de datos meteorológicos y climáticos existen varios paquetes de Python que contienen gran parte de los algoritmos de interpolación mencionados más arriba, como son *Iris* o *xESMF*. El segundo de los paquetes es el que se ha utilizado en este trabajo debido a su bajo gasto computacional y el eficiente resultado. Para mostrar un ejemplo, se ha utilizado el dato del modelo numérico WRF sobre el dominio delimitado por el radar de Palencia para la fecha del 20 de Octubre de 2019 a las 04:00 UTC, comprobando un gran aumento de la resolución espacial con el consiguiente suavizado de la mancha de precipitación (ver Figura 3.9).

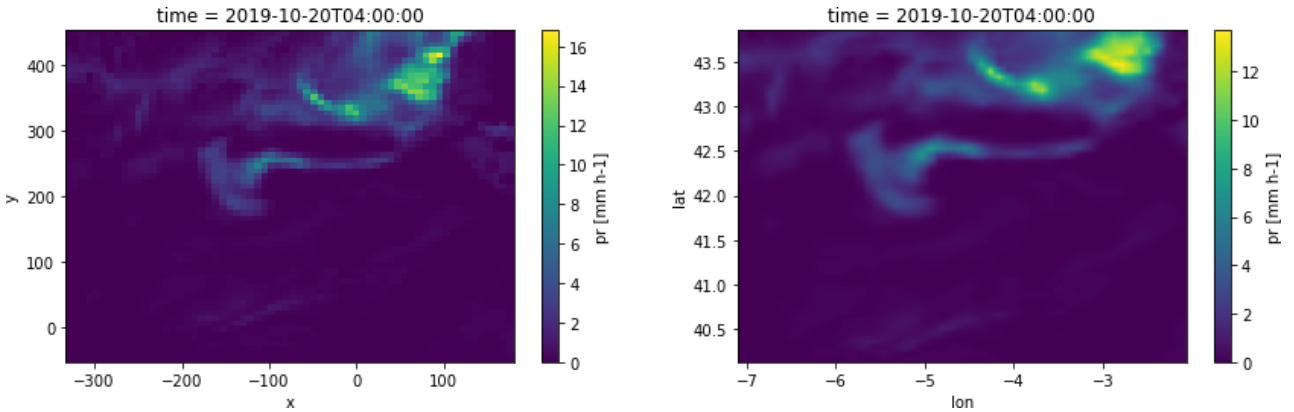


Figura 3.9: Datos antes (izquierda) y después (derecha) de realizar la interpolación bilineal, pasando las coordenadas curvilíneas ( $x$ ,  $y$ ) a coordenadas rectilíneas ( $\text{lon}$ ,  $\text{lat}$ ) en el dominio de las observaciones del radar regional situado en Palencia.

## **Parte III**

# **Técnicas: estructura, alimentación y análisis**



## Capítulo 4

# Modelización

### 4.1. Técnicas actuales (benchmarking)

Las aplicaciones de predicción a corto plazo usando técnicas estadísticas son numerosas y han sido utilizadas en regiones muy variadas, incluyendo en España [31]. Aunque son de gran interés, no se ha indagado sobre posibles mejoras en los algoritmos ya disponibles por defecto en el paquete de Python llamado *Pysteps*, sino que se ha añadido una explicación a grandes rasgos de las principales técnicas y funcionalidades usadas en este trabajo.

En términos generales la persistencia lagrangiana consiste en la posibilidad de explicar la evolución temporal del campo de precipitación por la advección de la última observación, donde advección se define como la variación de un escalar en un punto por el efecto de un campo vectorial. En este caso, dicho campo vectorial se denomina *campo de advección*.

Por tanto, el algoritmo de persistencia lagrangiana se divide en dos pasos fundamentales:

- Cálculo del campo de advección, normalmente mediante técnicas diferenciales de rastreo del campo de precipitación. Las técnicas de flujo óptico resaltan con respecto al resto, y una de ellas es especialmente usada durante el desarrollo de este trabajo: el método de Lukas-Kanade [32].
- Extrapolación de la última observación de radar mediante su uso junto con el campo calculado en el paso anterior.

Para el episodio de extrapolación existen diferentes métodos en función de las transformaciones que se aplican sobre las imágenes de radar sobre las que se realiza la advección. Estos métodos definen diferentes modelos que han sido implementados a lo largo de los años y que han sido acogidos por *Pyteps*, entre los que se han usado: Extrapolación, S-PROG y STEPS.

#### 4.1.1. Extrapolación

El método de extrapolación más sencillo es conocido como *vector-constante*, cuyo resultado se obtiene de la translación lineal y constante de cada uno de los píxeles correspondientes a la última observación de radar mediante el uso del campo de advección. El principal problema de esta translación lineal es que no permite la rotación de las predicciones.

El esquema semi-Lagrangiano nace a partir de la necesidad de dicha rotación de las predicciones, permitiéndolas mediante el seguimiento de las líneas de corriente que forman el campo de desplazamiento. En esta metodología, se divide el tiempo de desplazamiento en intervalos más pequeños y se calcula la advección del campo de precipitación para dichos tiempos. La extrapolación final se corresponde con la suma vectorial de los desplazamientos obtenidos en cada subintervalo mencionado.

*Pysteps* pone en nuestras manos la posibilidad de utilizar este esquema semi-Lagrangiano de extrapolación (ver Figura 4.1). Utilizando las dos primeras observaciones (del 2019-11-22 06:00 a 06:10, ambas inclusive) calcula el vector de desplazamiento con el que se extrapola la última observación (correspondiente al 2019-11-22 06:20) el número de pasos de tiempo que se le indique (30 minutos en este ejemplo).

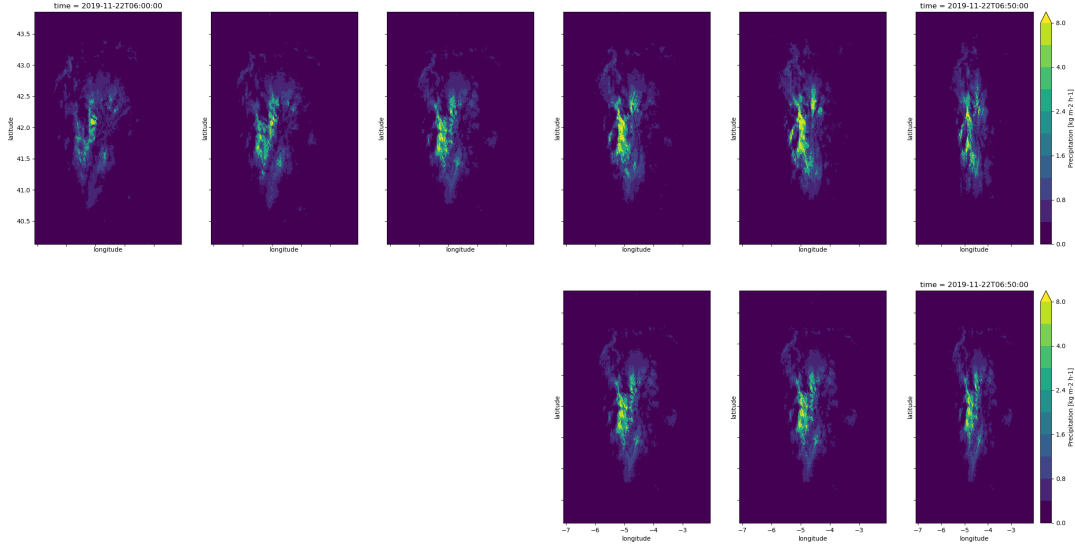


Figura 4.1: Observaciones procesadas del radar de Palencia para el período del 2019-11-22 06:00 al 2019-11-22 07:00 (arriba) y predicción (abajo) obtenida mediante el uso de la función de extrapolación proporcionada por *Pysteps*.

#### 4.1.2. S-PROG

S-PROG proviene de la abreviación del término inglés *Spectral Prognosis*, y es un modelo basado en la advección del campo de precipitación.

De acuerdo a Seed A.W. [33], el modelo está constituido por tres componentes diferentes:

1. Estimación del campo de desplazamiento mediante el uso de algoritmos de reconocimiento y rastreo de patrones
2. Descomposición del campo calculado en sus componentes de Fourier. Consiste en separar el ratio de precipitación en cascadas (ilustradas en este ejemplo de *Pysteps*) multiplicativas con diferentes escalas, así un método de cascada consiste en un número de campos que representan entre ellos un único análisis de lluvia.
3. Un modelo Lagrangiano para la advección del campo de precipitación que dependa de la escala sobre la que se trabaja. Esto se consigue utilizando un modelo auto-regresivo (lag 2, haciendo referencia a que estudia la correlación entre la muestra actual y la de dos tiempos anteriores [34]) que produce que el campo de previsión se suavice a medida que las estructuras detectadas evolucionan a través de sus ciclos de vida (ver Figura 4.2 para comprobar dicho suavizado en la predicción).

La motivación detrás del enfoque de S-PROG es separar las características a gran y pequeña escala en el patrón de precipitación, lo que permite que las trazas de precipitación se traten de manera independiente según su escala. Así, las características de escala más pequeña tendrán un tiempo de vida más corto (ver resultado en Figura 4.2), mientras que los patrones con escalas mayores persistirán en el tiempo.

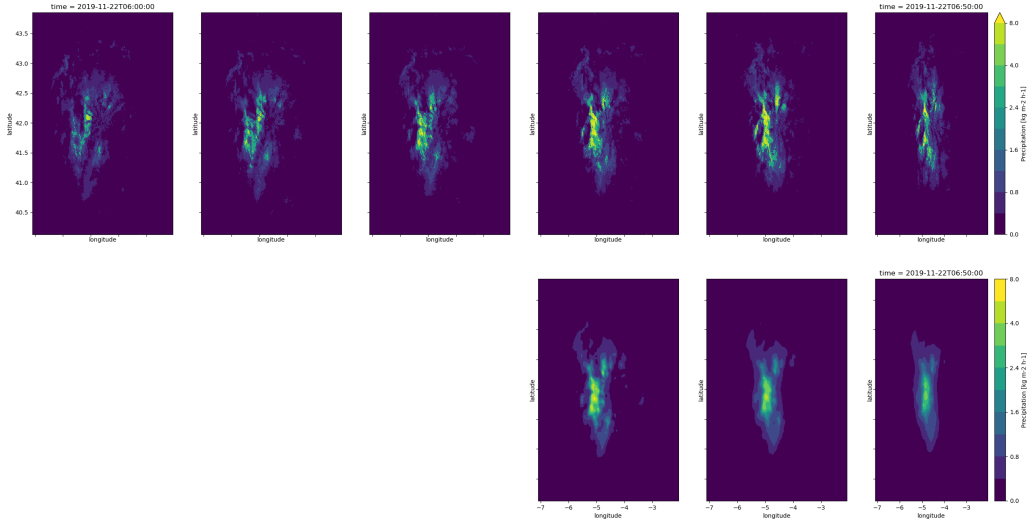


Figura 4.2: Observaciones procesadas del radar de Palencia para el período del 2019-11-22 06:00 al 2019-11-22 07:00 (arriba) y predicción (abajo) obtenida mediante el uso de la función S-PROG proporcionada por *Pysteps*.

#### 4.1.3. STEPS

En STEPS (del inglés *short-term ensemble prediction system*), se produce un conjunto con varios miembros, es decir, es lo que suele llamarse un método de *ensembles*.

Su enfoque está basado en la ampliación del modelo S-PROG para incluir un término estocástico que representa la varianza asociada al desarrollo impredecible de la precipitación [35]. En otras palabras, es extrapolado por un campo de velocidades calculado mediante el uso de un componente aleatorio que describe la incertidumbre de la advección. Para cada miembro del conjunto se define una cascada aleatoria diferente y se calcula un campo de velocidades de advección diferente. En dicho caso, el promedio de todas las extrapolaciones llevadas a cabo suavizará la incertidumbre asociada a la mancha de precipitación (ver Figura 4.3).

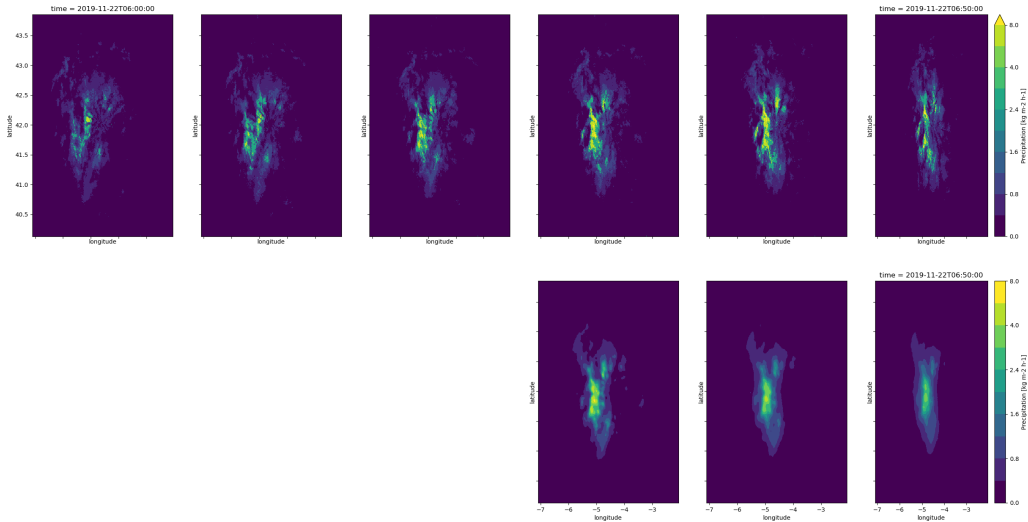


Figura 4.3: Observaciones procesadas del radar de Palencia para el período del 2019-11-22 06:00 al 2019-11-22 07:00 (arriba) y predicción (abajo), promedio de un conjunto de 20 miembros, obtenida mediante el uso de la función STEPS proporcionada por *Pysteps*.

Estas características comentadas (número de niveles de cascada, el número de conjuntos a producir para realizar el promedio, ...) son totalmente parametrizables mediante el uso del paquete *Pysteps*.

## 4.2. Modelos de aprendizaje profundo

### 4.2.1. Introducción al aprendizaje profundo

El *aprendizaje profundo* o *Deep Learning* se trata de una rama del aprendizaje automático que nace con el objetivo de realizar estudios dando un enfoque similar al que se da en el sistema nervioso de gran parte de los mamíferos, es decir, una red de neuronas que permita el flujo de información entre cada una de las neuronas que conforman dicho sistema nervioso. A esta red de neuronas se le refiere como *red neuronal*.

La neurona es la unidad base de la red neuronal. A cada neurona dentro de la red le llegan unos datos de entrada con los que realiza una suma ponderada cuyos pesos son ajustables. Posteriormente, se añade un sesgo que se ajusta también usando el mismo algoritmo de optimización. Como se puede comprobar, hasta ahora la salida de la neurona se corresponde con una suma de funciones lineales que da lugar a otra función lineal. En este preciso instante, entra en juego la función diseñada para captar las características no lineales de los datos, más conocida como *función de activación* (ver Figure 4.4).

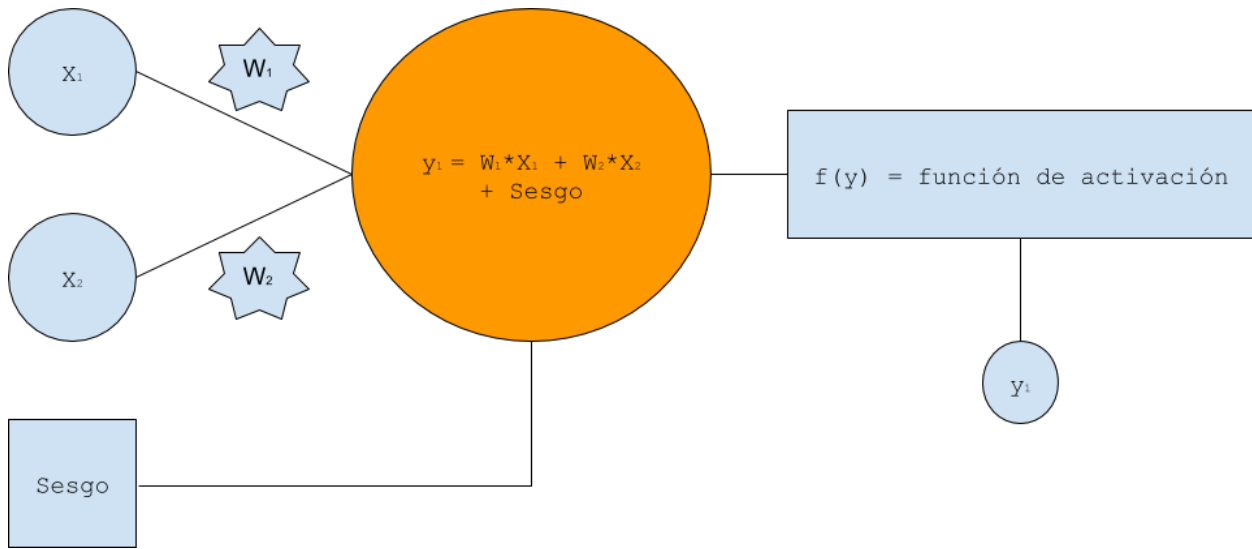


Figura 4.4: Representación esquemática de la recepción de datos, pesos y sesgo de una neurona cualquiera dentro de la red neuronal, incluyendo la transformación (lineal o no lineal) de la suma ponderada mediante el uso de la función de activación. En este caso, el número de datos de entrada que recibe la neurona de estudio es dos.

Las funciones de activación son diversas y se deben de usar acorde a los objetivos de la red. En este trabajo se han considerado tres tipos de funciones de activación distintas:

- **Lineal:** es una función lineal comúnmente conocida que toma las entradas, multiplicadas por los pesos de cada neurona, y crea una señal de salida proporcional a la entrada (ver Ecuación 4.1).

$$f(x) = cx \quad (4.1)$$

- **Relu (Rectified linear unit):** se trata de una función no lineal a trozos que tiene como principal ventaja, entre otras cosas, un ahorro computacional elevado. Esta función toma los valores de entrada y los transforma de tal forma que anula los valores negativos y dejan los positivos tal y como entran (ver Ecuación 4.2).

$$f(x) = \begin{cases} 0, & \text{si } x < 0 \\ x, & \text{si } x \geq 0 \end{cases} \quad (4.2)$$

Esta función de activación está expuesta al conocido problema de desvanecimiento del gradiente. La transformación a 0 de las sumas ponderadas negativas que llegan a una neurona a veces puede llevar a la neurona hacia estados en los que se vuelven inactivas para esencialmente todas las entradas, produciendo que no haya gradientes que fluyan hacia atrás a través de la neurona, y así la neurona se queda atascada en un estado de inactividad perpetua y "muere".

- Leaky Relu: se trata de una función de activación no lineal que nace como una variante de la ya explicada Relu con el objetivo de solventar potenciales problemas como el explicado anteriormente. Básicamente su funcionamiento se basa en transformar los valores introducidos multiplicando los negativos por un coeficiente rectificativo y dejando los positivos según entran (ver Ecuación 4.3).

$$f(x) = \begin{cases} \alpha x, & \text{si } x < 0 \\ x, & \text{si } x \geq 0 \end{cases} \quad (4.3)$$

Una vez se han definido las funciones de activación que se han utilizado, cabe destacar que las neuronas se pueden agrupar formando estructuras cada vez más complejas. Una de estas estructuras es la *capa*, definiéndose así a cada agrupación de neuronas que reciben los mismos datos de entrada (procedentes de cada una de las neuronas que conforman la capa anterior en la red neuronal) y los transforman de la misma forma mediante el uso de una función de activación común. Una vez transformados los datos, son enviados a la siguiente agrupación de neuronas o capa. Existen varios tipos de capas en una red neuronal: la capa de entrada y la capa de salida de la red, y las capas intermedias conocidas como ocultas (ver Figura 4.5).

Dentro de una capa, el número de neuronas es definido al inicio, al igual que el método de inicialización de los pesos de la capa. Además, el número de capas que conforma una red neuronal es muy variable y puede verse aumentado enormemente dependiendo de la problemática a considerar, es por esto que este tipo de aprendizaje sea "*profundo*".

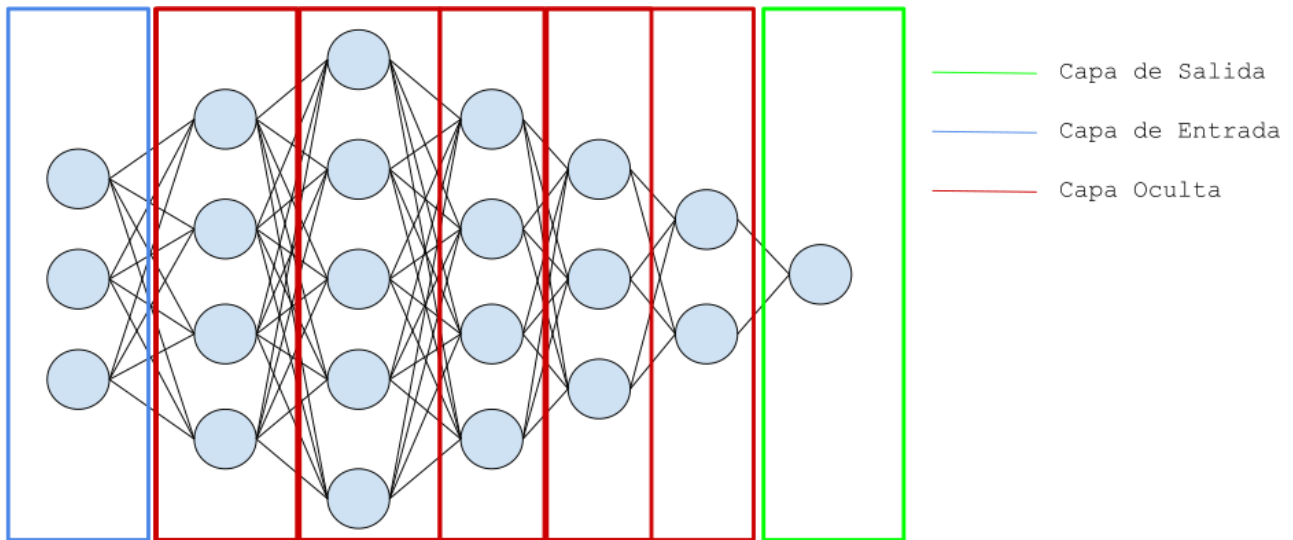


Figura 4.5: Representación esquemática de una red neuronal

Hasta el momento, se ha hablado únicamente de capas ocultas tratando de generalizar. Sin embargo, dentro de la tipología de capas existe una gran variedad: densa, convoluciones (Conv), long-short term memory (LSTM), Conv-LSTM, de aplanamiento, de regularización, para reducir y aumentar la dimensionalidad por el método de máximos, ... Durante la explicación de la estructura de las redes desarrolladas para este trabajo se explicarán brevemente algunas de estas capas.

Después de esta breve introducción a los elementos y definiciones básicas del deep learning, se estudiará el posible valor añadido de las técnicas de aprendizaje profundo sobre el problema de la predicción de precipitación a muy corto plazo mediante el desarrollo de dos modelos: *Raindictia* y *RaindictiaWRF*.

## 4.2.2. Métodos y estructuras empleados

### 4.2.2.1. Estructura Encoder-Decoder

Tanto *Raindictia* como *RaindictiaWRF* tienen un esquema de capas similar y son modelos de aprendizaje profundo estructurados siguiendo la referencia dada por la familia U-Net [21] (se recomienda [este video](#) explicativo acerca de la estructura) y *SegNet* [22]. RainNet, preparada para este tipo de problemática, presenta también una arquitectura similar.

Las funciones utilizadas para la activación de las diferentes capas de las arquitecturas presentadas por *Raindictia* y *RaindictiaWRF* se pueden resumir en los tres tipos ya definidos más arriba: *lineal*, *Relu* (*Rectified linear unit*) y *Leaky Relu*.

Siguiendo en la línea de los modelos mencionados más arriba, *Raindictia* y *RaindictiaWRF* presentan una arquitectura *encoder-decoder* [36] (ver Figura 4.6). El primer elemento se basa en una estructura de capas que produce una reducción de la dimensionalidad espacial de los datos de entrada acompañada de convoluciones. Una vez los datos se encuentran en un estado intermedio gracias a las transformaciones realizadas por el *encoder*, el segundo elemento realiza la restitución a la dimensionalidad de entrada, es decir, presenta un conjunto de capas que aumenta la dimensionalidad espacial hasta el estado original ayudándose de convoluciones.



Figura 4.6: Esquema general de la estructura *Encoder-Decoder* utilizada en la modelización de la red neuronal.

Entre capas del *encoder* y del *decoder* se realizan conexiones conocidas como *conexiones de salto entre capas* [37], las cuales se han utilizado con el objetivo de mejorar el flujo de información entre elementos de la red neuronal.

En términos generales, *Raindictia* y *RaindictiaWRF* están formadas, en distinto número, por: capas de convolución en 2D, capas de reducción de la dimensionalidad, con sus respectivas capas de incremento de la dimensionalidad para retornar a las condiciones originales, saltos de conexión entre el *encoder* y el *decoder*, y capas de regularización.

### 4.2.2.2. Capa de Convolución

Las capas de convolución 2-dimensionales se basan en el uso de un filtro o *kernel* de tamaño elegible ( $N \times N$ ) que recorre la matriz de datos de entrada. Este tipo de capa es altamente utilizada en el procesamiento de imágenes porque se mantiene invariante a translaciones, es decir, si la capa reconoce un patrón en un lugar característico de una imagen, luego será capaz de reconocerlo en cualquier otra parte de la imagen. En otras palabras, el uso de capas de convolución en una red neuronal permite el aprendizaje de patrones locales por parte de la red.

Todas las convoluciones realizadas en *Raindictia* y *RaindictiaWRF* hacen uso de un filtro  $3 \times 3$  cuyos pesos son inicializados con una distribución normal truncada centrada en 0 con desviación estándar igual a la raíz cuadrada de 2 dividido por el número de unidades de entrada en el tensor de pesos.

Además dichas convoluciones se pueden realizar de dos maneras diferentes: haciendo un relleno de la matriz de entrada para obtener otra matriz del mismo tamaño después de pasar el filtro (en el paquete de Python *Tensorflow* se indica mediante un parámetro de la función `Conv2D`, `padding = same`) (ver Figura 4.7), o pasando el filtro directamente sobre la matriz de entrada sin rellenarla (`padding = valid`). Normalmente, se usa el parámetro `padding` como `same` para evitar problemas de reducción excesiva de la matriz de datos y la infrarepresentación de los píxeles situados en esquinas y bordes en el espacio de características (conocido como *efectos de borde*).

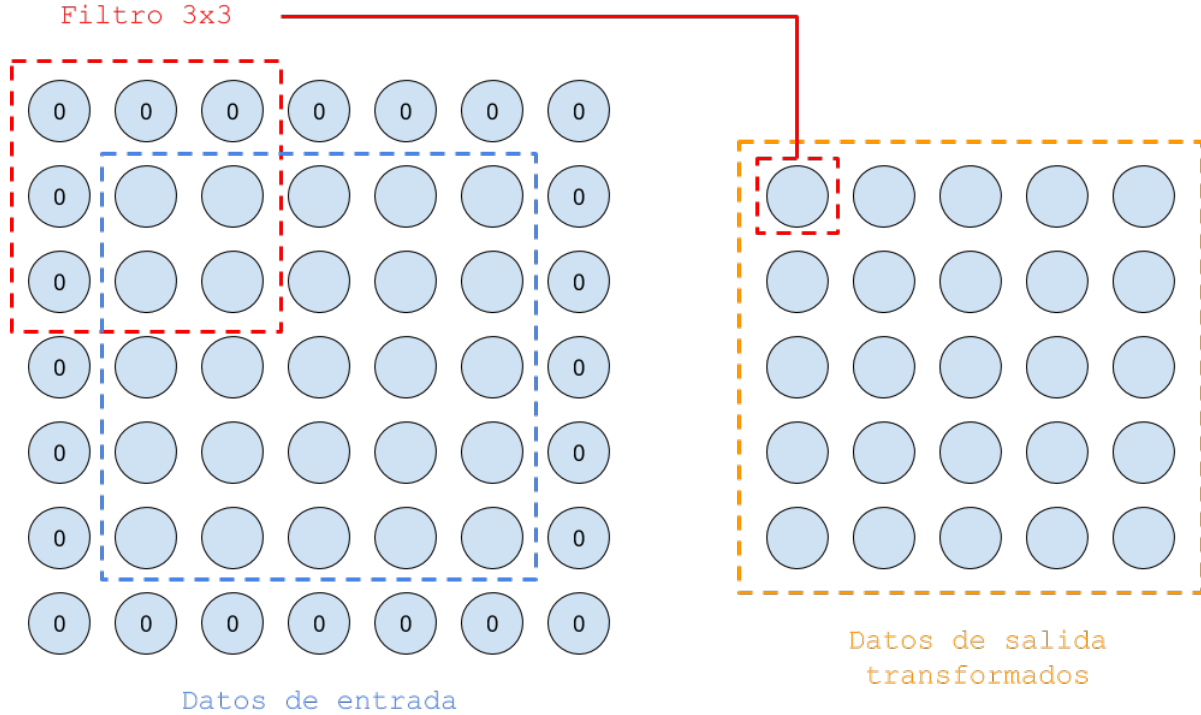


Figura 4.7: Representación de una convolución 2-dimensional con un filtro de tamaño 3x3 sobre una matriz de datos de entrada de tamaño 5x5. La matriz de salida también es 5x5 debido al relleno de los datos de entrada con 0s en los bordes.

El tamaño final una vez aplicado el filtro se puede calcular de la siguiente forma. Teniendo una matriz de neuronas de entrada  $N \times N$ , si se hace uso de un filtro de tamaño  $F \times F$  para recorrer la matriz de entrada, el tamaño final de los datos transformados es:

$$N \times N * F \times F \longrightarrow (N + 2P - F + 1) \times (N + 2P - F + 1) \quad (4.4)$$

- Si no se realiza un relleno de los datos (`padding = valid`), entonces  $P = 0$ .
- Si se realiza un relleno de los datos (`padding = same`) para obtener el mismo tamaño en la matriz de los datos de salida que en la matriz de entrada, entonces  $P = (F - 1) / 2$ .

#### 4.2.2.3. Capa de reducción por máximos

Las capas de reducción por máximos se utilizan comúnmente sobre los datos transformados una vez se ha aplicado un filtro de convolución, es decir, consecutivamente a una capa de convolución. Su finalidad es reducir el coste computacional del entrenamiento mediante una reducción de las dimensiones espaciales de los datos de entrada. Esta reducción del coste la consiguen debido, entre otras cosas, a que no son parametrizables.

Como su propio nombre indica, se trata de un capa que reduce los datos usando un factor de escalada elegible (del mismo modo que el tamaño de filtro de las convoluciones). Este factor de



escalada para el eje X e Y (2-dimensional, aunque también se puede utilizar esta capa con factores 3-dimensionales) produce la división del eje X y el eje Y en cuadrados (si el factor es igual en el eje X que en el Y, rectángulos en caso contrario), de tal manera que elige el máximo valor de los datos contenidos en cada división de los datos de entrada.

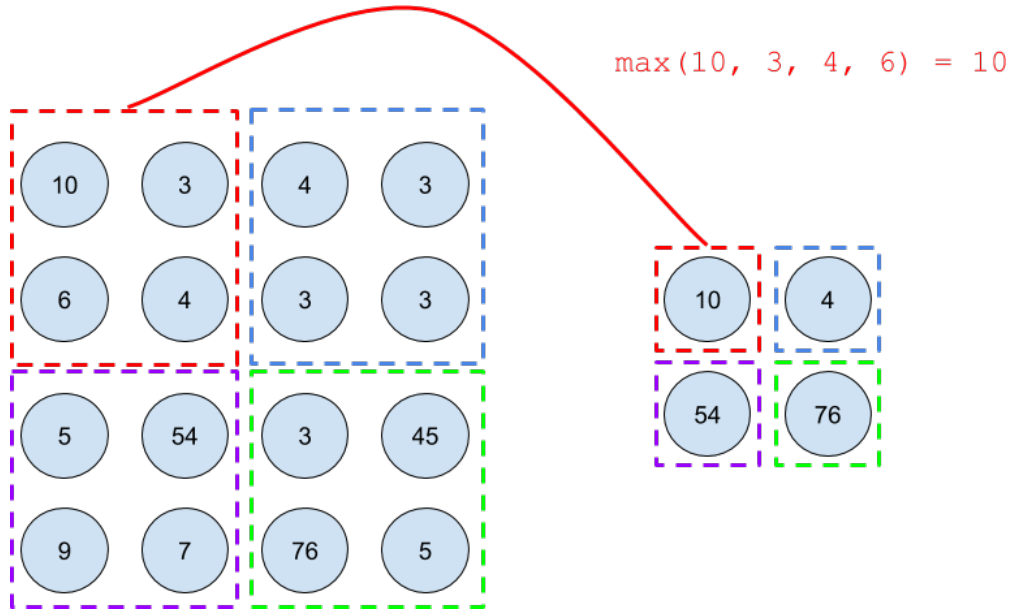


Figura 4.8: Representación esquemática de la transformación realizada por una capa de reducción por máximos con factor de escalado (2, 2) sobre una matriz de datos de entrada de tamaño 4x4.

En el ejemplo de Raindictia y RaindictiaWRF, se han utilizado capas de reducción por máximos 2-dimensionales con un factor de escalado, en su mayoría, de (2, 2) (en ocasiones (4, 4)) (ver Figura 4.8).

De acuerdo a la documentación [21], las redes neuronales de la familia U-Net (como puede ser Raindictia y RaindictiaWRF) deben de tener un número de capas de reducción por máximos ( $n$ ) tal que  $2^{n+1}$  sea directamente proporcional a la dimensionalidad de los datos de entrada. En el caso de este trabajo, se cumple.

Existen otro tipo de capas de reducción que siguen la misma metodología, pero utilizando la media o la norma en lugar del máximo.

#### 4.2.2.4. Capa de incremento de la dimensionalidad

La capa de incremento de la dimensionalidad tiene una funcionalidad similar a interpolación de los datos de WRF que se realizó en la sección *WRF: Acoplamiento espacio-temporal*, es decir, se quiere transformar una matriz de datos en baja resolución (mayor tamaño de malla) a una matriz con una resolución espacial más alta. En este caso, se hace uso de la función por defecto del paquete de python *Keras*, con lo que en lugar de realizar una interpolación bilineal se utiliza un esquema basado en el vecino más cercano.

#### 4.2.2.5. Capa de apagado de neuronas

La capa de apagado de neuronas (también conocida como *dropout*) [38] se basa en la desactivación de un ratio (%) de neuronas aleatorias de una capa de una red neuronal (ver Figura 4.9). El objetivo de esta capa es reducir la cantidad de sobre-ajuste de la red neuronal al conjunto de datos de entrenamiento de manera que ninguna neurona memorice parte de la entrada.



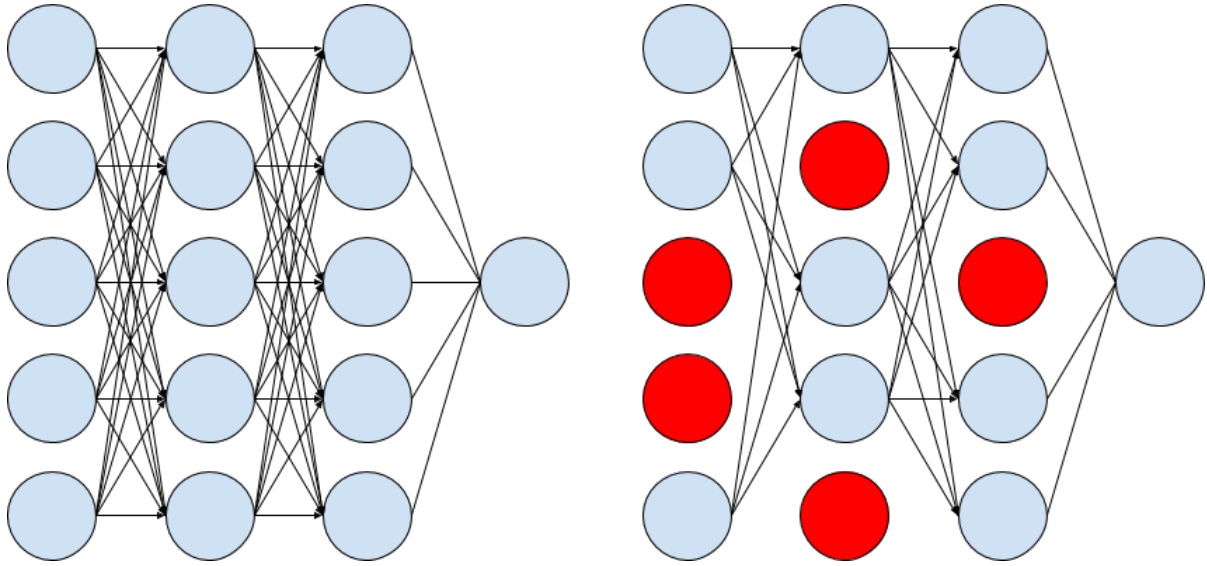


Figura 4.9: Representación esquemática de las consecuencias de aplicar capas de apagado de neuronas, donde cada columna representa una capa oculta dentro de una red neuronal. A la izquierda, las capas antes de realizar el apagado de neuronas, mientras que a la derecha se encuentran en rojo las neuronas que han sido desactivadas por la capa de apagado con un ratio de apagado de neuronas igual a 0.4, 0.4 y 0.2, respectivamente.

Al realizarse un apagado aleatorio de las neuronas, esta técnica se puede aproximar a la creación de un modelo en cada iteración dando lugar a un modelo por conjuntos (como es el caso de STEPS).

#### 4.2.2.6. Conexiones entre capas no contiguas

Las conexiones de salto entre capas no son una capa en sí dentro de la red neuronal, sino que se basan en la concatenación de los datos de salida de una capa del encoder con los datos de entrada de una capa del decoder, con el objetivo de permitir el flujo de información desde un componente a otro (ver Figura 4.10).

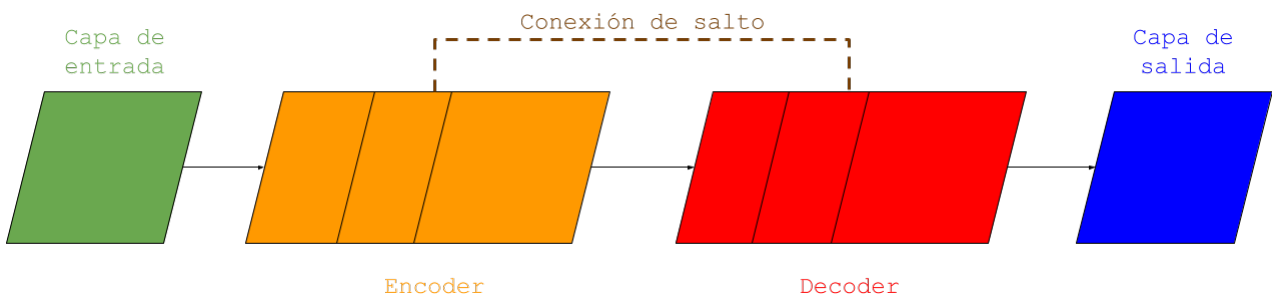


Figura 4.10: Representación del salto de conexiones entre una capa del encoder y otra del decoder para permitir el flujo de información entre ambos.

#### 4.2.3. Arquitectura de los modelos diseñados

Una vez explicados todas los métodos y tipos de estructuras que han sido utilizadas, se presentan Raindictia (ver Figura 4.11) y RaindictiaWRF (ver Figura 4.12).

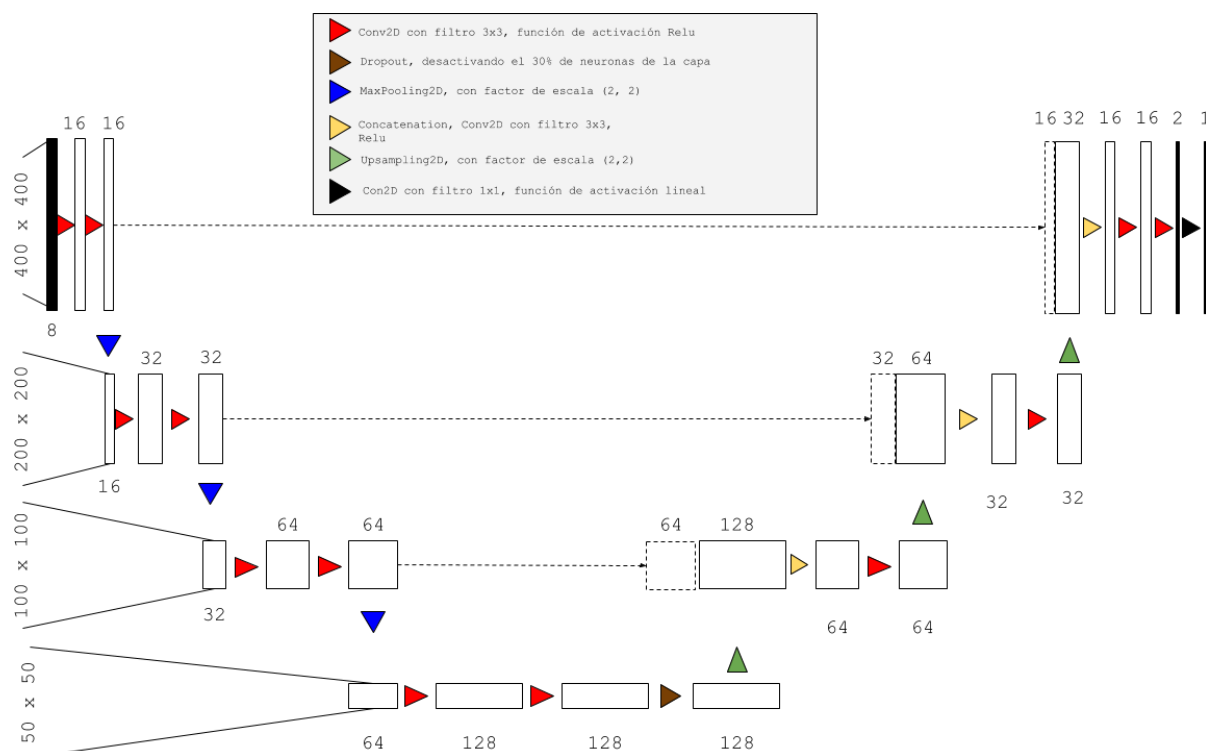


Figura 4.11: Esquema de capas del modelo Raindictia.

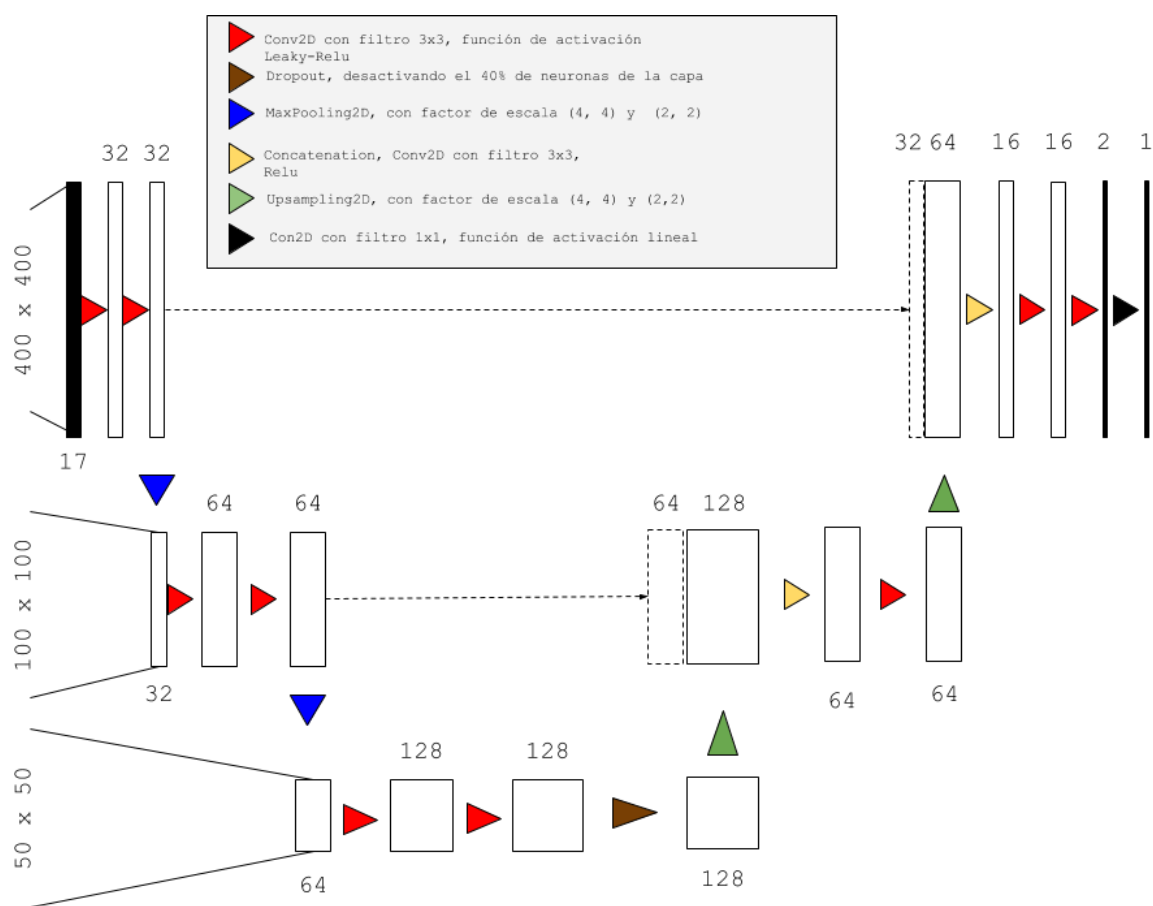


Figura 4.12: Esquema de capas del modelo RaindictiaWRF.

#### 4.2.4. Entrenamiento

En términos generales, se debe notar que el entrenamiento de una red neuronal sobre un conjunto de datos es un problema de optimización de la función  $f(x, y)$  que representa a la red neuronal. Este proceso de optimización es un proceso iterativo, donde a cada iteración se le suele referir como época.

Para cada muestra de entrenamiento  $(x, y)$  que se modeliza dentro de una misma época, se pasa las características predictoras de la muestra  $(x)$  como datos de entrada para alimentar el modelo. Entonces, el modelo realiza todas las transformaciones correspondientes a las capas por las que está constituido. Una vez los datos de entrada han pasado por el modelo, se obtiene una estimación ( $y_{pred} = f(x)$ ) del valor real. Esta estimación es utilizada para conocer el error entre la predicción ( $y_{pred}$ ) y el valor real conocido ( $y$ ) dado por la función  $L(y_{pred}, y)$ , conocida como función de coste o función de pérdida. Esta función es de un valor trascendental y se deberá de elegir posterior a estudiar las características del modelo utilizado, como se verá más adelante.

Una vez obtenido este error se utiliza el algoritmo de *backpropagation* para asignar a cada neurona la fracción del error total que se debe a ella. Posteriormente, cada neurona actualiza sus pesos utilizando un algoritmo de optimización que se ajuste a la problemática considerada (RMSprop, adam, adamax, ...). Uno de los parámetros de este algoritmo de optimización es el ratio de aprendizaje, también conocido como *learning rate*.

El proceso mencionado es realizado para cada una de las muestras de entrenamiento  $(x, y)$  dentro de una época.

##### 4.2.4.1. Experimento: muestras y set-up

Como se ha explicado, se tienen datasets divididos en periodos de 15 días. Estos datasets tienen una gran dimensionalidad debido a su alta resolución espacio-temporal, lo que dificulta la carga de datos en memoria para un periodo completo de entrenamiento. Para poder cargar todo el periodo del que se disponen datos de una única vez, es decir, sin necesidad de entrenar el modelo de nuevo para considerar nuevos datos, se han producido todas las muestras posibles a partir de los datasets iniciales.

Las características esenciales de estas muestras son:

- Cada muestra  $(x, y)$  tiene unas dimensiones espacio-temporales de tamaño  $(8, 400, 400)$  y  $(1, 400, 400)$ , respectivamente.
- No debe existir más de un 5 % de valores nulos (0s) en la matriz que contiene los datos de entrada a la red.
- La mancha de precipitación no debe ser constante en dos tiempos consecutivos, es decir, el valor de la mancha en  $t - 40\text{min}$  no puede ser igual al valor en  $t - 30\text{min}$ .
- Se realiza una transformación de las muestras del tipo [19]:

$$x' = \ln(x + 0,01) \text{ e } y' = \ln(y + 0,01).$$

Las muestras se han definido de tal manera para evitar ciertos problemas asociados con las posibles paralizaciones de las imágenes de radar ofrecidas por AEMET y el uso de datos que no están balanceados.

Este procedimiento se ha llevado a cabo en cada uno de los radares regionales que se ha deseado modelizar: A Coruña, Palencia y Asturias. Esta decisión tiene como objetivo el uso de datos de radar que se encuentran tanto en la costa como en el interior de la Península Ibérica. Aunque los radares están muy cercanos, el número de muestras válidas que se ha podido extraer de cada uno

es muy variable (ver Tabla 4.1).

Número de muestras	A Coruña	Asturias	Palencia
Entrenamiento	7368	4766	5526
Validacion (in-train)	1300	841	975

Tabla 4.1: Número de muestras para las distintas etapas de modelización de Raindictia (entrenamiento y validación en el entrenamiento) en función del radar de estudio.

Estos datos son diferentes en el caso del entrenamiento de RaindictiaWRF debido a que únicamente se precisan de datos de radar para los tiempos entre el 16 de Octubre de 2019 y el 16 de Mayo de 2020. Por tanto, todas las muestras de la Tabla 4.1 con la variable tiempo fuera del rango mencionado no se han tenido en cuenta para el entrenamiento del modelo RaindictiaWRF quedando un número inferior al mostrado anteriormente (ver Tabla 4.2).

Número de muestras	A Coruña	Asturias	Palencia
Entrenamiento	6458	3848	3960
Validacion (in-train)	1140	680	699

Tabla 4.2: Número de muestras para las distintas etapas de modelización de RaindictiaWRF (entrenamiento y validación en el entrenamiento) en función del radar de estudio.

Debido a que tampoco ha sido posible cargar todas las muestras en su conjunto en la memoria de la máquina sobre la que se ha trabajado, ha sido necesario seguir una estrategia por lotes. En otras palabras, en lugar de alimentar a la red con todas las muestras de entrenamiento, estas muestras de entrenamiento se han dividido en lotes de menor tamaño formados de manera aleatoria. En este caso, el modelo no se actualiza con cada muestra sino con cada lote de muestras. Los lotes no presentan un tamaño elevado debido al dispositivo sobre el que se ha realizado el experimento.

Para entrenar el modelo se han utilizado los paquetes de *Tensorflow* y *Keras* instalados en Python para realizar la ejecución sobre una GPU (Graphical Processing unit) Nvidia GTX 1060Ti de 4Gb de memoria.

#### 4.2.4.2. Función de coste

La función de coste o función de pérdida se define como un escalar que trata de determinar el error entre el valor estimado por la red neuronal y el valor real, como ya se ha dicho anteriormente. Esta función tiene que tratar de captar las características y complejidades del problema de tal forma que sea robusta y eficiente dentro del algoritmo de optimización.

En el problema de predicción de precipitación a muy corto plazo se tienen unos datos que no están balanceados. Como se ha podido ver gran parte del campo de precipitación de la imagen de radar de las muestras (x, y) son ceros, por tanto una mala elección de la función de coste llevaría al modelo a realizar una predicción en la que todo (o casi todo) el campo de precipitación fuera nulo.

Guiado en cierta medida por la función de coste que utilizó Rainnet se comenzó utilizando el logaritmo del coseno hiperbólico que se presenta como beneficioso en la optimización de la estructura encoder-decoder porque en lugar de introducir únicamente la penalización de tipo  $L_2$  como el error cuadrático medio (RMSE), esta variante introduce también una penalización  $L_1$  para los valores más grandes [39].

Sin embargo, a la hora de utilizar dicha función de coste los valores seguían siendo bajos y la suma total de precipitación en la predicción se quedaba muy lejos de la suma total real. Por tanto,

necesitábamos una función de coste que penalizara más a aquellos errores en los que la predicción del modelo se quedará por debajo del valor real. Por ello hemos hecho uso de una regresión cuantílica que toma la forma de la Ecuación 4.5.

$$L(e_i|\alpha) = \begin{cases} \alpha e_i, & \text{si } e_i \geq 0 \\ (\alpha - 1)e_i, & \text{si } e_i < 0 \end{cases} \quad (4.5)$$

donde  $e_i = y_i - f(x_i)$  es la desviación de la predicción con respecto a la realidad a nivel de píxel, y  $\alpha$  es el cuantil elegido para el proceso de optimización.

La elección del cuantil depende en gran medida de los datos de los que se dispone. Es de gran acierto decir que si se elige un cuantil  $\alpha = 0.5$  simplemente se está optimizando con el error medio absoluto (MAE) (ver Figura 4.13. La elección del cuantil en el caso de Raindictia y RaindictiaWRF ha sido optimizada mediante un proceso iterativo basado en el uso del método de bisección.

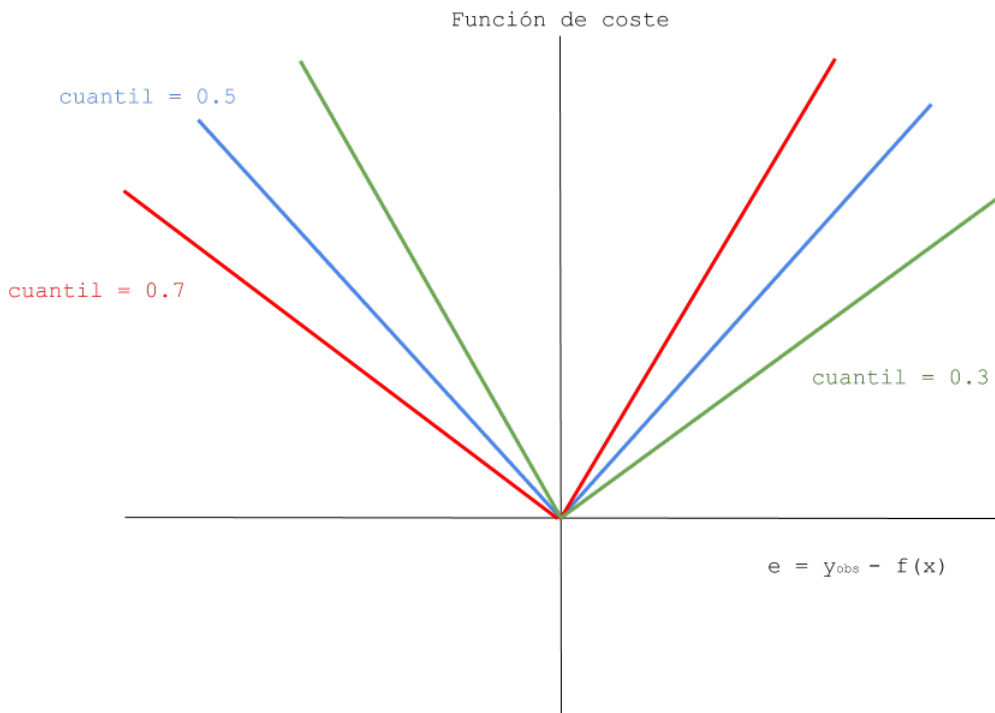


Figura 4.13: Visualización de la función de coste dada por la Ecuación 4.5 utilizando varios valores del cuantil.

#### 4.2.4.3. Algoritmo de optimización

Como ya se ha explicado anteriormente, una vez asociado a cada neurona (parámetro) el coste correspondiente dentro de una iteración se realiza la optimización de los parámetros asociados a dicha neurona con el objetivo de reducir la función de coste final.

El algoritmo de optimización utilizado en el caso de Raindictia y RaindictiaWRF es el Adam (adaptive moment estimation) [40].

#### Ratio de aprendizaje

El ratio de aprendizaje o *learning rate* es un parámetro de optimización en el proceso de entrenamiento de una red neuronal. Se trata de un parámetro de especial importancia ya que se encarga de la actualización de los pesos en cada una de las épocas del entrenamiento de la red. Normalmente, es una característica de estudio en la parametrización de una red neuronal debido a que la elección

de un parámetro de aprendizaje demasiado pequeño elevará el tiempo de computo para llegar a alcanzar el mínimo de la función de coste, mientras que la elección de un parámetro demasiado grande podrá concurrir en el problema de *explosión del gradiente* haciendo que el modelo se vuelva inestable.

En el proceso de entrenamiento de Raindictia y RaindictiaWRF se ha decidido hacer uso de un ratio de aprendizaje que no permanezca constante a lo largo de las iteraciones. En otras palabras, se ha utilizado un learning rate con un esquema de actualización cuyo objetivo es que una vez el entrenamiento se ha acercado al mínimo de la función de coste, únicamente se busquen pequeños movimientos de optimización.

#### 4.2.5. Evaluación

Para la evaluación de los modelos entrenados se ha hecho uso de muestras totalmente independientes al proceso de entrenamiento producidas de igual forma que las que se han comentado en la sección *Experimento: muestras y set-up*. Las muestras han sido generadas con datos entre el 16 de Octubre de 2019 y el 16 de Mayo de 2020 con el objetivo de realizar la evaluación de todas las técnicas y modelos con los mismos datos. En la Tabla 4.3 aparecen las cantidades de muestras utilizadas para la evaluación de las técnicas y modelos.

Número de muestras	A Coruña	Asturias	Palencia
Evaluación	1200	1000	1200

Tabla 4.3: Número de muestras para las distintas etapas de evaluación de los modelos Raindictia y RaindictiaWRF en función del radar de estudio.

##### 4.2.5.1. Predicción iterativa

Para cada una de la muestras señaladas, se ha realizado una predicción para los siguientes 60 minutos. Es decir, alimentando a nuestro modelo con los datos de  $t-70\text{min}$ , ...,  $t-10\text{min}$ ,  $t$ , se ha realizado una predicción para  $t+10\text{min}$ , ...,  $t+60\text{min}$ .

Esto no es posible con el modelo tal y como ha sido diseñado debido a que la capa de salida del modelo tiene dimensiones (1, 400, 400), o lo que es lo mismo, los modelos creados tienen la capacidad de predecir  $t+10\text{min}$ . Con el objetivo de predecir más allá, es necesario concatenar la predicción del modelo como última observación y posteriormente utilizar el modelo para hacer una nueva predicción.

Una vez se tienen los resultados para cada una de las iteraciones se realiza la transformación inversa de los datos para volver a unidades de mm/h:  $y = \exp(y') - 0,01$ .

Esta técnica de predicción tiene como principal inconveniente el arrastrar errores procedentes de las primeras predicciones. Esto quiere decir que para la predicción de  $t + 10$  minutos se van a utilizar únicamente imágenes de radar, sin embargo para la predicción  $t + 60$  minutos solamente dos de los pasos de tiempo con los que se alimenta el modelo ( $t-70\text{min}$  y  $t-60\text{min}$ ) van a ser observaciones verdaderas.

##### 4.2.5.2. Métricas

Para la evaluación se tienen dos matrices: una matriz de predicción con dimensiones (6, 400, 400) y una matriz de observación con dimensiones (6, 400, 400), donde ambas presentan sus datos de precipitación como una variable continua con unidades de mm/h.

Aunque los datos estén dados de forma continua se han considerado también métricas que consideran la precipitación como una variable discreta (ocurrencia o no de precipitación). Para ello, se

le ha pasado un límite a partir del cual se considera que ocurre la precipitación. Este límite se ha instaurado durante la validación en 0.125 mm/h.

El software desarrollado permite realizar una evaluación de las componentes espacial y temporal de la precipitación en los diferentes modelos que se diseñen y entrenen.

Entre las métricas consideradas para el proceso de evaluación se encuentran [41]:

- La raíz del error cuadrático medio (RMSE):

$$RMSE = \sqrt{\frac{1}{N} \sum_1^N (y_i - f(x_i))^2} \quad (4.6)$$

- El error absoluto medio (MAE):

$$MAE = \frac{1}{N} \sum_1^N |y_i - f(x_i)| \quad (4.7)$$

- El índice crítico de éxito (CSI):

$$CSI = \frac{TP}{TP + TN + FN} \quad (4.8)$$

donde los valores de TP, TN y FN se pueden obtener de la matriz de confusión mostrada en la Tabla 4.4.

	Observado >0.125 mm/h	Observado <0.125 mm/h
Predicho >0.125 mm/h	Verdadero Positivo (TP)	Falso Positivo (FP)
Predicho <0.125 mm/h	Falso Negativo (FN)	Verdadero Negativo (TN)

Tabla 4.4: Matriz de confusión para las métricas que tratan la precipitación como variable discreta.

- La correlación de Pearson (R):

$$R = correlation(y, f(x)) = \frac{cov(y, f(x))}{\sqrt{var(y)var(f(x))}} \quad (4.9)$$

## Capítulo 5

# Resultados y análisis

### 5.1. Evolución de los modelos durante la etapa de entrenamiento

La etapa de entrenamiento se ha llevado de igual forma para todos los modelos y radares considerados. Para ello, se ha elegido un número constante de épocas de entrenamiento (30) y se ha parametrizado la función de coste dada por la Ecuación 4.5 con un cuantil de 0.55. Además, el parámetro de aprendizaje inicial para el algoritmo de optimización es igual (0.001) y decrece de la misma forma (factor 1/2 cada 3 épocas de entrenamiento) en todos los casos.

Esto produce que la optimización de los modelos en el proceso de entrenamiento sea muy similar tanto para Raindictia (ver Figura 5.1) como para RaindictiaWRF (ver Figura 5.2). A nivel de radar, es posible comprobar que el radar de Palencia comienza en una franja inferior que a los otros dos. Esto no tiene una gran impacto ya que al final del entrenamiento el radar de Asturias y el de Palencia presentan valores de pérdida muy similares, a diferencia del radar de A Coruña que no supera la barrera del 0.10.

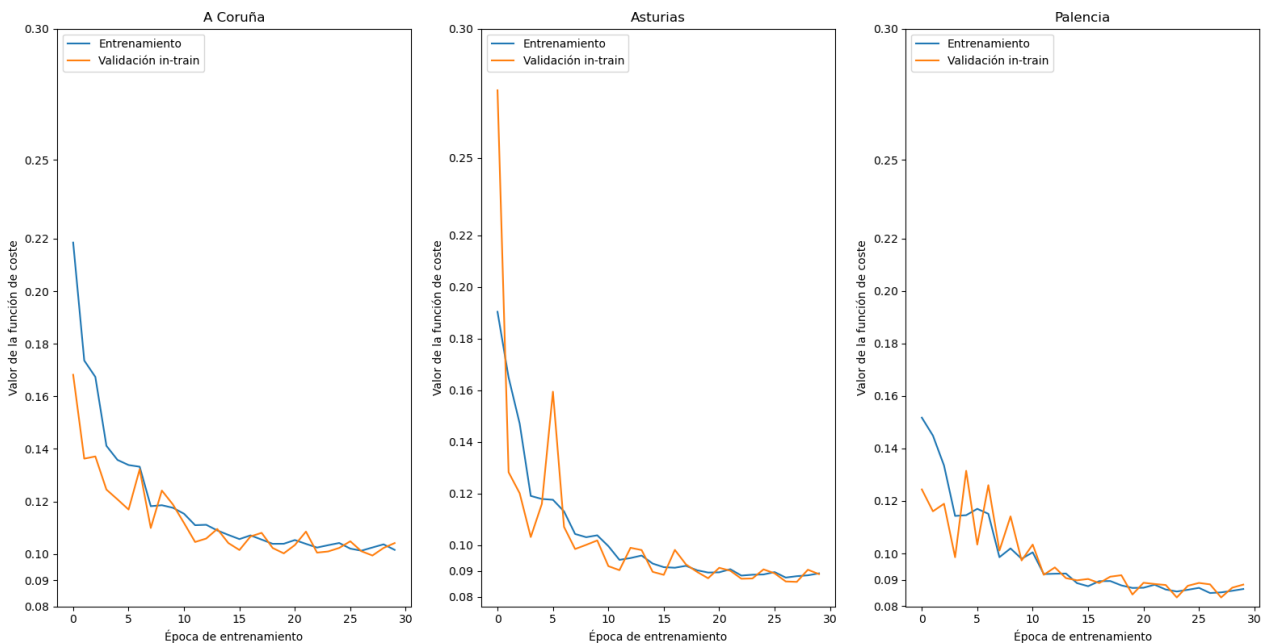


Figura 5.1: Valor de la función de pérdida en función de la época de entrenamiento para el modelo Raindictia, tanto para la modelización de los datos de entrenamiento (85 %) como para los de validación in-train (15 %).



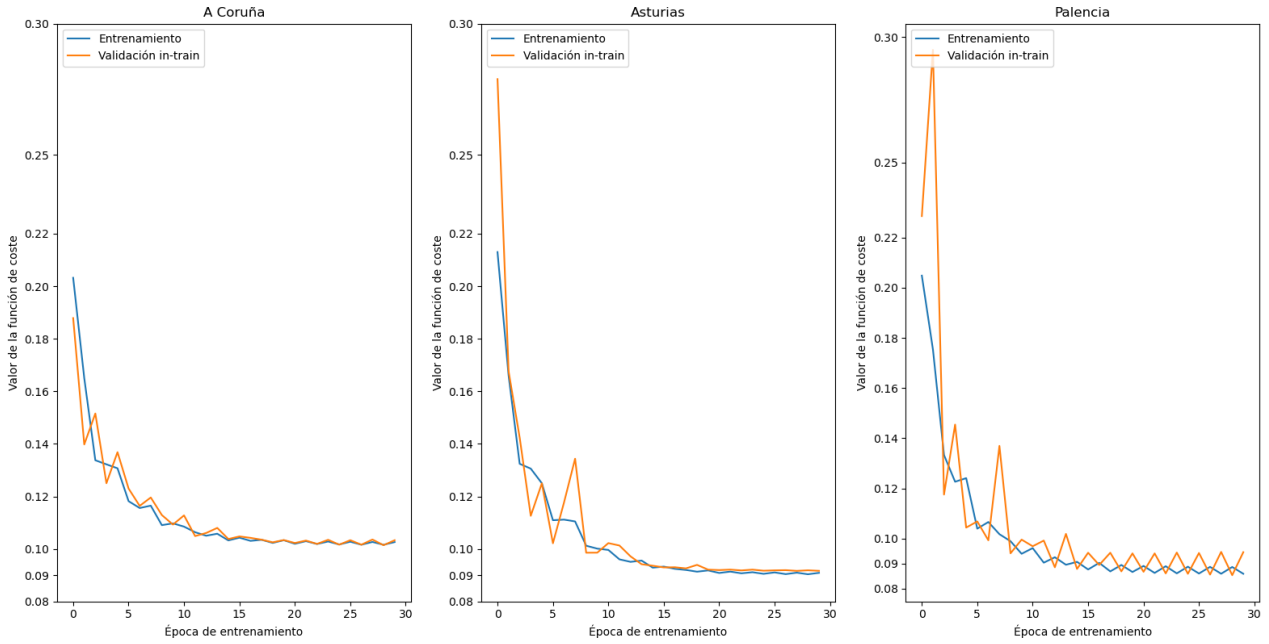


Figura 5.2: Valor de la función de pérdida en función de la época de entrenamiento para el modelo RaindictiaWRF, tanto para la modelización de los datos de entrenamiento (85 %) como para los de validación in-train (15 %).

Una de las principales características que puede llegar a apreciarse prestando algo más de atención es el decaimiento del ratio de aprendizaje que permite a ambos modelos en el caso de Asturias y Palencia superar la barrera de pérdida igual a 0.10, lo cual no ocurre en el caso de A Coruña. Otra curiosidad es la inicialización con error menor de Raindictia para el radar de Palencia, esto puede deberse a una inicialización en los parámetros de las neuronas que favoreció a la optimización del modelo.

Viendo que los modelos siguen un proceso de entrenamiento similar para cada uno de los radares regionales incluidos en el estudio, se pudiera haber incluido una simulación en la que se tuvieran en consideración todas y cada una de las muestras creadas (independientemente del radar).

Es importante hacer notar que aunque los modelos tengan el mismo número de parámetros, la matriz de entrada de RaindictiaWRF es más del doble de grande que la matriz de entrada del modelo Raindictia. Por tanto, el tiempo de entrenamiento también se duplica prácticamente, pasando de 5 minutos a 12 minutos por época de entrenamiento entre los dos modelos.

## 5.2. Evaluación sobre periodo de tiempo

Una vez comentadas las características esenciales de la etapa de entrenamiento, es hora de pasar a la evaluación de nuestros modelos y compararlos principalmente con las técnicas estadísticas comentadas en la sección *Técnicas basadas en la persistencia lagrangiana*. También usaremos la persistencia euleriana como comparación, lo que consiste en coger la observación real en el último tiempo ( $t$ ) y utilizarla para utilizarla como predicción los diferentes horizontes temporales de estudio.

Aunque las muestras de evaluación se hayan escogido de manera aleatoria y no presenten continuidad temporal, se puede considerar como un periodo temporal en su conjunto. Tomando como ejemplo el caso de A Coruña, se tienen 1200 muestras de validación conformadas por un predictor ( $x$ ) y un predictando ( $y$ ), donde en su conjunto suman 140 minutos reales de datos de precipitación con frecuencia diezminutal. Sumando el total de minutos que componen las muestras de validación se obtiene una equivalencia total de en torno a 8 días.

El software de evaluación de los modelos se ha desarrollado de tal manera que para cada muestra se ha obtenido una tabla similar a la Tabla 5.1. Una vez se han calculado los resultados correspondientes a cada una de las muestras se ha realizado el promedio muestral de las métricas para cada uno de los tiempos de predicción ( $t+10\text{min}$ , ...,  $t+60\text{min}$ ). En otras palabras, la Tabla 5.1 no es más que el promedio de 1200 tablas correspondientes a las muestras de A Coruña. Este mismo procedimiento se ha llevado a cabo con los otros radares regionales tomados en consideración para el proceso de entrenamiento y evaluación.

En las tablas 5.1, 5.2 y 5.3 se encuentran los resultados de las métricas de evaluación como función del tiempo de predicción y de la técnica utilizada. Dichas tablas vienen a estudiar el error espacial en cada uno de los pasos de tiempo de la predicción para cada uno de los métodos. Cabe destacar que no se han detectado anomalías en los datos producidos durante el periodo de validación, excepto para la estructura RaindictiaWRF entrenada sobre los datos de entrenamiento de A Coruña. Estas anomalías se deben a inestabilidades en el modelo con ciertas muestras (400 del total de 1200) para horizontes de predicción superiores a 30 minutos. Se hablará más adelante sobre ello.

Prácticamente en el caso de todos los radares en los que se ha basado el estudio se aprecia un rendimiento similar entre Raindictia y RaindictiaWRF, superando a las técnicas estadísticas basadas en la persistencia lagrangiana y en especial a la persistencia euleriana. Este rendimiento similar entre ambos modelos de aprendizaje profundo era esperado debido a que la evaluación se ha realizado sobre una hora de predicción y muy probablemente no sea lo suficiente para visualizar el carácter predictivo del modelo de mesoescala. Es decir, en la predicción actual las observaciones de radar influyen en gran medida a la red neuronal, sería para pasos de tiempo mayores donde los datos proporcionados por WRF comenzarían a tomar parte en el juego.

		Persistencia	Extrapolación	S-PROG	STEPS	Raindictia	RaindictiaWRF
RMSE	t+10min	0.42	0.34	0.27	0.29	0.30	0.32
	t+20min	0.47	0.40	0.31	0.33	0.33	0.35
	t+30min	0.49	0.43	0.33	0.35	0.35	0.36
	t+40min	0.50	0.45	0.34	0.36	0.36	0.38
	t+50min	0.50	0.47	0.35	0.37	0.37	0.39
	t+60min	0.51	0.48	0.36	0.37	0.37	0.41
MAE	t+10min	0.093	0.065	0.063	0.075	0.056	0.064
	t+20min	0.110	0.084	0.079	0.089	0.066	0.074
	t+30min	0.118	0.095	0.088	0.097	0.072	0.082
	t+40min	0.124	0.104	0.095	0.102	0.076	0.088
	t+50min	0.128	0.110	0.101	0.106	0.080	0.093
	t+60min	0.131	0.115	0.105	0.108	0.083	0.098
CSI (0.125 mm/h)	t+10min	0.46	0.61	0.62	0.52	0.64	0.64
	t+20min	0.36	0.50	0.52	0.43	0.53	0.55
	t+30min	0.31	0.43	0.46	0.37	0.45	0.47
	t+40min	0.28	0.37	0.41	0.32	0.38	0.41
	t+50min	0.26	0.33	0.37	0.28	0.33	0.37
	t+60min	0.24	0.29	0.33	0.24	0.29	0.32
Correlación	t+10min	0.43	0.61	0.72	0.52	0.69	0.68
	t+20min	0.32	0.48	0.61	0.43	0.57	0.60
	t+30min	0.27	0.40	0.54	0.37	0.49	0.52
	t+40min	0.24	0.34	0.48	0.32	0.42	0.45
	t+50min	0.22	0.29	0.43	0.28	0.36	0.39
	t+60min	0.20	0.25	0.38	0.25	0.31	0.34

Tabla 5.1: Comparación de las métricas utilizadas para los diferentes métodos tanto de referencia (Persistencia, Extrapolación, S-Prog y STEPS) como de los modelos desarrollados (Raindictia y RaindictiaWRF) para el radar regional de A Coruña.

		Persistencia	Extrapolación	S-PROG	STEPS	Raindictia	RaindictiaWRF
RMSE	t+10min	0.33	0.28	0.22	0.24	0.23	0.23
	t+20min	0.36	0.32	0.25	0.27	0.25	0.25
	t+30min	0.37	0.35	0.26	0.28	0.26	0.27
	t+40min	0.38	0.36	0.27	0.29	0.27	0.28
	t+50min	0.39	0.38	0.28	0.29	0.28	0.28
	t+60min	0.39	0.38	0.28	0.29	0.29	0.29
MAE	t+10min	0.065	0.048	0.047	0.057	0.039	0.040
	t+20min	0.077	0.062	0.059	0.066	0.046	0.047
	t+30min	0.083	0.071	0.066	0.072	0.051	0.052
	t+40min	0.087	0.077	0.071	0.076	0.055	0.055
	t+50min	0.089	0.082	0.075	0.078	0.058	0.058
	t+60min	0.091	0.085	0.077	0.079	0.060	0.060
CSI (0.125 mm/h)	t+10min	0.47	0.61	0.62	0.51	0.63	0.62
	t+20min	0.37	0.49	0.52	0.43	0.53	0.53
	t+30min	0.32	0.42	0.45	0.37	0.45	0.45
	t+40min	0.29	0.36	0.40	0.32	0.39	0.39
	t+50min	0.27	0.32	0.36	0.29	0.34	0.33
	t+60min	0.25	0.28	0.32	0.26	0.30	0.29
Correlación	t+10min	0.46	0.59	0.70	0.66	0.67	0.67
	t+20min	0.35	0.46	0.60	0.54	0.57	0.57
	t+30min	0.30	0.37	0.52	0.45	0.49	0.49
	t+40min	0.27	0.31	0.46	0.39	0.43	0.43
	t+50min	0.24	0.27	0.40	0.34	0.37	0.37
	t+60min	0.23	0.23	0.36	0.30	0.33	0.32

Tabla 5.2: Comparación de las métricas utilizadas para los diferentes métodos tanto de referencia (Persistencia, Extrapolación, S-Prog y STEPS) como de los modelos desarrollados (Raindictia y RaindictiaWRF) para el radar regional de Asturias.

		Persistencia	Extrapolación	S-PROG	STEPS	Raindictia	RaindictiaWRF
RMSE	t+10min	0.34	0.29	0.23	0.24	0.24	0.24
	t+20min	0.38	0.34	0.26	0.28	0.27	0.27
	t+30min	0.41	0.37	0.28	0.30	0.28	0.29
	t+40min	0.42	0.39	0.29	0.31	0.30	0.30
	t+50min	0.43	0.40	0.30	0.31	0.31	0.31
	t+60min	0.43	0.41	0.31	0.32	0.32	0.32
MAE	t+10min	0.064	0.048	0.047	0.057	0.041	0.042
	t+20min	0.080	0.063	0.060	0.069	0.049	0.050
	t+30min	0.088	0.073	0.068	0.076	0.056	0.056
	t+40min	0.094	0.080	0.074	0.080	0.061	0.060
	t+50min	0.098	0.085	0.079	0.083	0.065	0.064
	t+60min	0.101	0.089	0.082	0.085	0.069	0.067
CSI (0.125 mm/h)	t+10min	0.51	0.63	0.64	0.53	0.66	0.65
	t+20min	0.40	0.52	0.54	0.45	0.57	0.55
	t+30min	0.34	0.44	0.47	0.38	0.49	0.47
	t+40min	0.30	0.39	0.41	0.33	0.43	0.41
	t+50min	0.27	0.34	0.37	0.29	0.38	0.36
	t+60min	0.25	0.30	0.33	0.26	0.34	0.32
Correlación	t+10min	0.51	0.62	0.73	0.69	0.70	0.71
	t+20min	0.38	0.49	0.62	0.57	0.60	0.60
	t+30min	0.31	0.40	0.54	0.49	0.52	0.52
	t+40min	0.27	0.33	0.48	0.42	0.45	0.45
	t+50min	0.24	0.29	0.43	0.36	0.39	0.39
	t+60min	0.22	0.25	0.38	0.31	0.35	0.34

Tabla 5.3: Comparación de las métricas utilizadas para los diferentes métodos tanto de referencia (Persistencia, Extrapolación, S-Prog y STEPS) como de los modelos desarrollados (Raindictia y RaindictiaWRF) para el radar regional de Palencia.

Haciendo hincapié en los resultados mostrados en las tablas 5.1, 5.2 y 5.3, se aprecia que la única referencia competitiva en el marco de los modelos de aprendizaje profundo es S-PROG. Este modelo presenta unas correlaciones altas, sin embargo parece que el suavizado que realiza a las características menos relevantes del campo de precipitación lo penaliza en métricas como el CSI. Esto se debe a que el rastro de precipitaciones de menor escala tiene un menor tiempo de vida en este tipo de modelos. Sin embargo, en Raindictia y RaindictiaWRF se ha conseguido la no pérdida del patrón espacial mediante el uso de la regresión cuántica como función de coste. También por la función de coste utilizada que intenta penalizar la no predicción de los valores de precipitación de mayor tamaño (asociado normalmente con tormentas) se produce el aumento en métricas como el RMSE, que en cambio, no parece verse tan afectado en las técnicas estadísticas como S-PROG y STEPS.

Después de haber hecho una valoración general de los resultados consideramos apropiado pasar a una evaluación individualizada a nivel de radar, donde podemos comparar la actuación de nuestros modelos en las diferentes zonas del territorio español peninsular.

En el caso de la predicciones sobre las muestras de test del radar de A Coruña (ver Figura 5.3) se comprueba que entre Raindictia y RaindictiaWRF hay diferencias a considerar. Esto se debe a que el modelo RaindictiaWRF es inestable en el caso de A Coruña. Puede deberse a los datos de evaluación o a los datos utilizados durante el entrenamiento del modelo. Lo cierto es que de las 1200 muestras de validación, únicamente 800 han podido ser utilizadas debido a la alta inestabilidad para horizontes de predicción de más de 30 minutos. Por dar un ejemplo, existen muestras que una vez modelizadas emitían un RMSE superior a  $10e+10$ ... Esto se debe tener en cuenta, y es especialmente extraño ya que como veremos a continuación los modelos que usan WRF entrenados sobre los otros radares no dan este tipo de resultado.

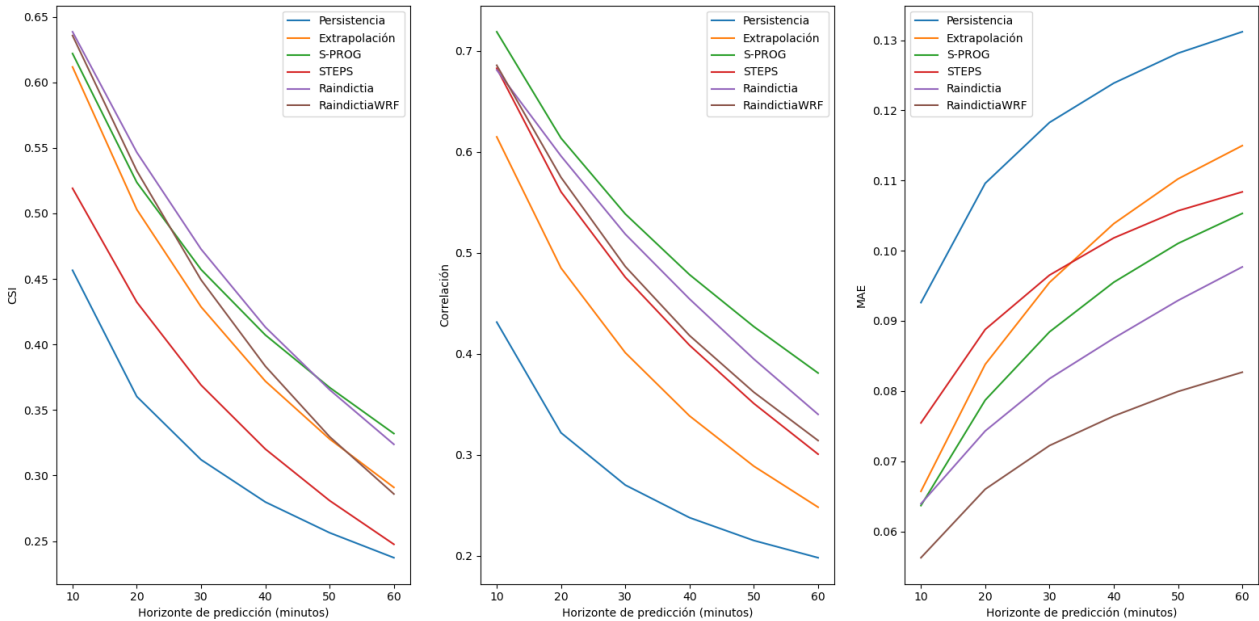


Figura 5.3: Representación de las métricas CSI, Correlación y MAE en función del horizonte de predicción para cada una de las técnicas utilizadas. Los resultados se corresponden a la aplicación de los modelos sobre el conjunto de muestras de evaluación del radar de A Coruña.

Por otro lado, para el radar de Asturias (ver Figura 5.4) se aprecia que Raindictia y RaindictiaWRF presentan valores de error muy semejantes, pareciendo RaindictiaWRF el más preciso. Sin embargo, se encuentra muy a la par de la técnica estadística S-PROG que, si bien no tiene un índice de éxito crítico elevado para los primeros pasos de tiempo, tiene una mejor actuación en esta métrica para horizontes de predicción de más de 40 minutos que la que pueden presentar los modelos de aprendizaje profundo.

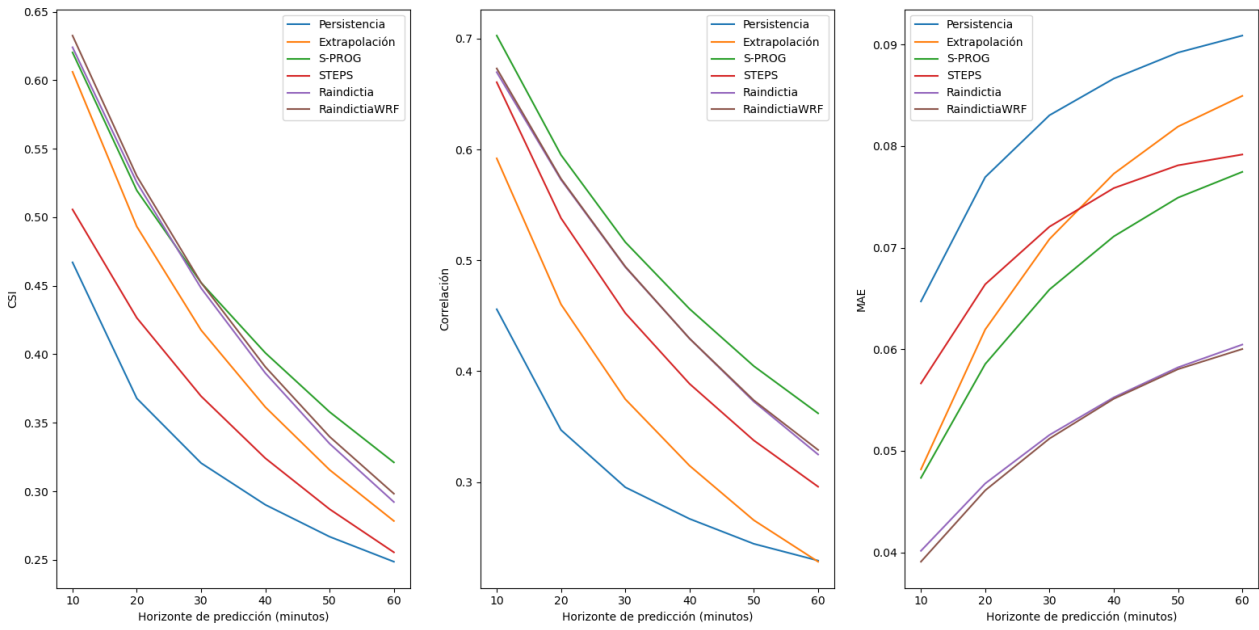


Figura 5.4: Representación de las métricas CSI, Correlación y MAE en función del horizonte de predicción para cada una de las técnicas utilizadas. Los resultados se corresponden a la aplicación de los modelos sobre el conjunto de muestras de evaluación del radar de Asturias.

Sin embargo, RaindictiaWRF es superior a S-PROG en todos los horizontes de predicción para el caso del radar regional situado en Palencia, aunque el modelo de aprendizaje profundo Raindictia sigue quedándose un peldaño por debajo en predicción a horizontes de evaluación más elevados (ver Figura 5.5). En el caso de la correlación ocurre lo mismo que para los radares situados en A Coruña y Asturias, el modelo basado en la persistencia lagrangiana S-PROG supera en todos los horizontes de predicción a los modelos de aprendizaje profundo.

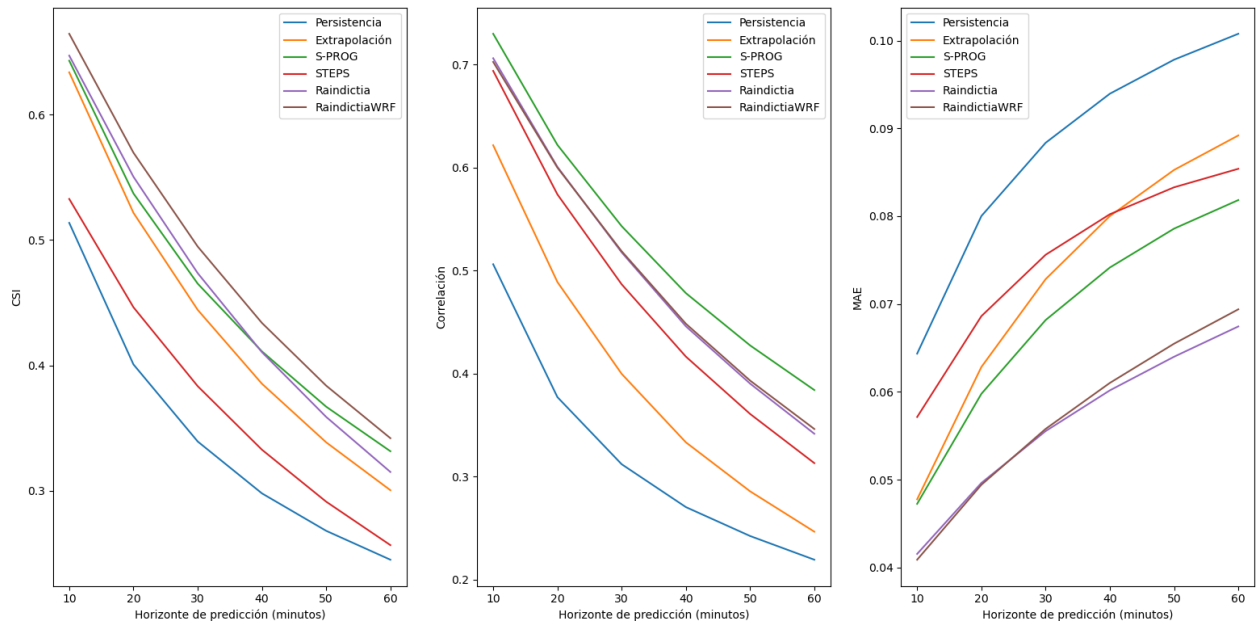


Figura 5.5: Representación de las métricas CSI, Correlación y MAE en función del horizonte de predicción para cada una de las técnicas utilizadas. Los resultados se corresponden a la aplicación de los modelos sobre el conjunto de muestras de evaluación del radar de Palencia.

En el caso de todos los radares regionales estudiados, los modelos de *deep learning* presentan un MAE muy por debajo del resto de técnicas y muy similar entre ambas estructuras de modelo. Esto puede deberse a la gran cantidad de ceros que constan en una observación de radar.

En muchas ocasiones, el ratio de lluvia es una variable pocas veces utilizadas y lo que realmente interesa es la acumulación de lluvia tanto espacialmente, como temporalmente. En primer lugar, apreciando el total de precipitación predicho, en términos espaciales, se puede observar que en casi todos los horizontes de predicción nos quedamos en promedio alejados del valor observado (ver Figuras 5.6 5.7 5.8) cuando se hace uso de los modelos de aprendizaje profundo. Sin embargo, las técnicas estadísticas parecen robustas en mostrar una propiedad semi-conservativa de la precipitación en los diferentes pasos de tiempo y casan muy bien con la distribución total de precipitación observada.

El que los modelos de aprendizaje profundo se alejen del valor acumulado de precipitación se puede deber a una elección baja del cuantil de la función de coste, produciendo que se pueda seguir bien el patrón espacial en las primeras iteraciones pero que se pierda la mancha de precipitación al aumentar el horizonte de predicción. El método para encontrar el cuantil elegido fue el método de bisección. Se ha podido comprobar durante la ejecución de los experimentos que la predicción era muy sensible al cuantil elegido debido al método iterativo de predicción. Es decir, se predecían valores altos de predicción en  $t+10\text{min}$ , los valores predichos en  $t+20\text{min}$  van a ser aún más altos... con lo que había que buscar un equilibrio.

En otras palabras, las técnicas estadísticas poseen la capacidad de trasladar el total de precipitación que ven en el tiempo inicial ( $t$ ) a los siguientes estados temporales ( $t+10\text{min}$ , ...,  $t+60\text{min}$ ,...). Sin embargo, se puede considerar el suavizado y pérdida del total de precipitación espacial como una característica intrínseca de los modelos de aprendizaje profundo. Raindictia y RaindictiaWRF han sido entrenados de manera que tenían que aprender a reducir el valor de una función de coste, lo cual inevitablemente ha introducido una pérdida de información en el primer paso de tiempo. Sin embargo, lo que sucede en otros pasos de tiempo no es una propiedad intrínseca de los modelos de *deep learning* sino que está asociado al esquema de predicción iterativo utilizado.

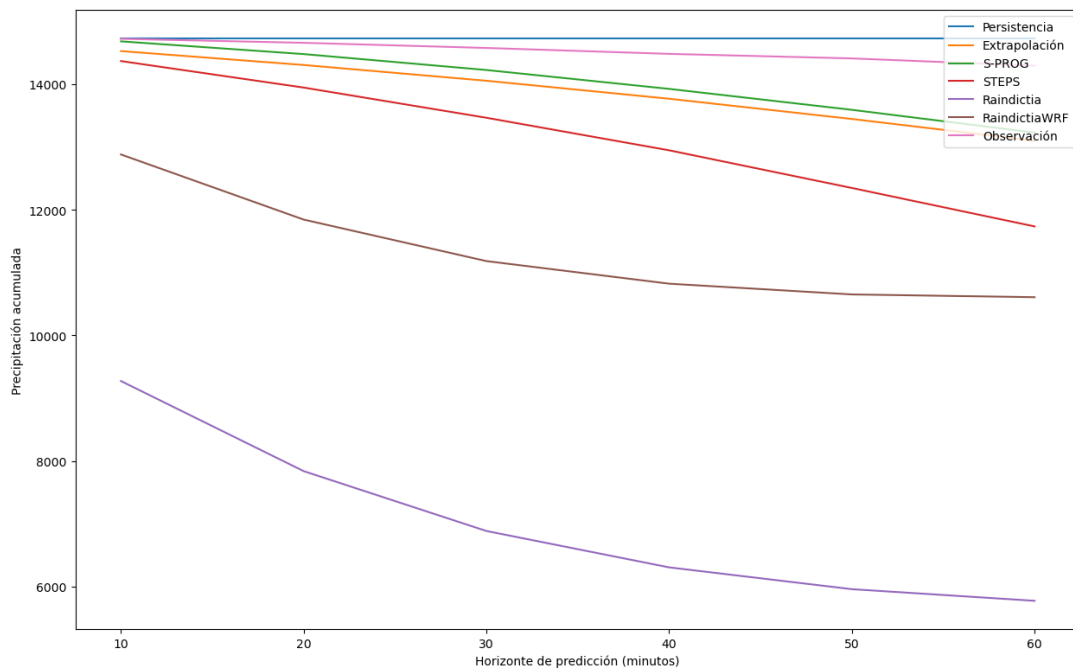


Figura 5.6: Representación de la precipitación espacial acumulada en función del horizonte de predicción para cada una de las técnicas utilizadas. Los resultados se corresponden a la aplicación de los modelos sobre el conjunto de muestras de evaluación del radar de A Coruña.

Justamente, de la Figura 5.6 se sacan conclusiones muy dispares. Un modelo más inestable como es RaindictiaWRF en el caso del radar regional de A Coruña presenta una menor desviación con respecto a los datos de la observación para la precipitación total acumulada. Sin embargo, en el caso de Raindictia (totalmente estable) se va desviando más y más notablemente al pasar por los diferentes horizontes de predicción.

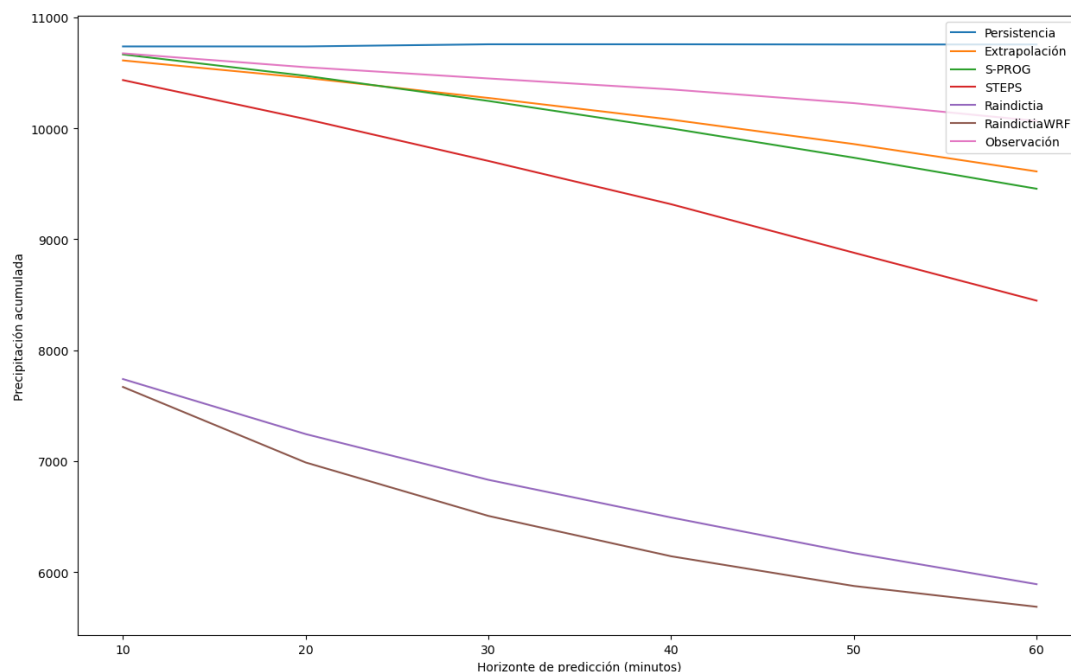


Figura 5.7: Representación de la precipitación espacial acumulada en función del horizonte de predicción para cada una de las técnicas utilizadas. Los resultados se corresponden a la aplicación de los modelos sobre el conjunto de muestras de evaluación del radar de Asturias.

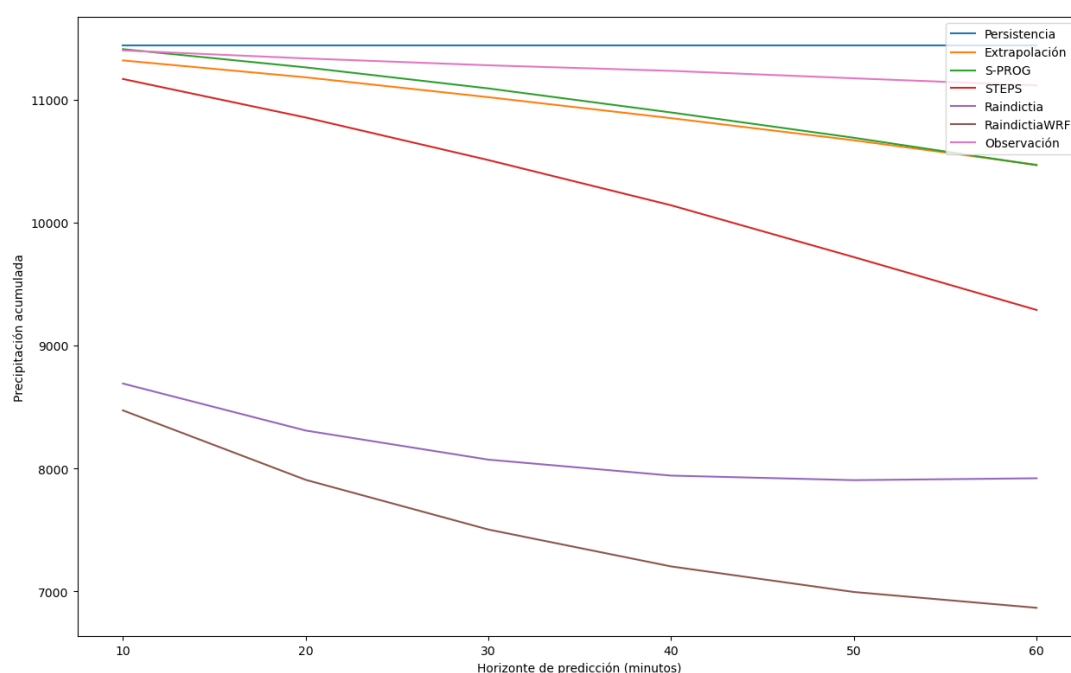


Figura 5.8: Representación de la precipitación espacial acumulada en función del horizonte de predicción para cada una de las técnicas utilizadas. Los resultados se corresponden a la aplicación de los modelos sobre el conjunto de muestras de evaluación del radar de Palencia.

Además de realizar una evaluación espacial para cada horizonte de predicción, se ha calculado el total de precipitación (mm/h) acumulado, durante los 60 minutos posteriores a la última observación, con el objetivo de comprobar la correlación entre el total de precipitación predicho y el total de precipitación observado. Atendiendo al diagrama de dispersión (ver Figuras 5.9 y 5.10) de los radares de Asturias y Palencia para los modelos de aprendizaje profundo Raindictia y RaindictiaWRF, se comprueba lo explicado anteriormente pero a nivel de precipitación acumulada durante todo el periodo de predicción (espacio-temporalmente).

Es decir, para la gran mayoría de las muestras se aprecia que la precipitación total acumulada durante la primera hora de predicción es inferior al valor acumulado de la observación. Si la precipitación acumulada predicha se ajustará realmente a la observación, la mayoría de valores (puntos rojos) deberían estar situados en torno a la diagonal de dispersión (línea azul). Se puede obtener la correlación entre la precipitación acumulada sobre la primera hora de predicción y la misma variable pero con los datos de observaciones. En el caso del modelo Raindictia para el radar de Asturias, la correlación entre dichas variables es de 0.899 y para el modelo de RaindictiaWRF se obtiene que es de 0.894. Para realizar una comparación con las de técnicas lagrangianas, el modelo S-PROG presenta una correlación entre las variables mencionadas igual a 0.927, mientras que el método de extrapolación más común produce un valor cercano a 0.938.

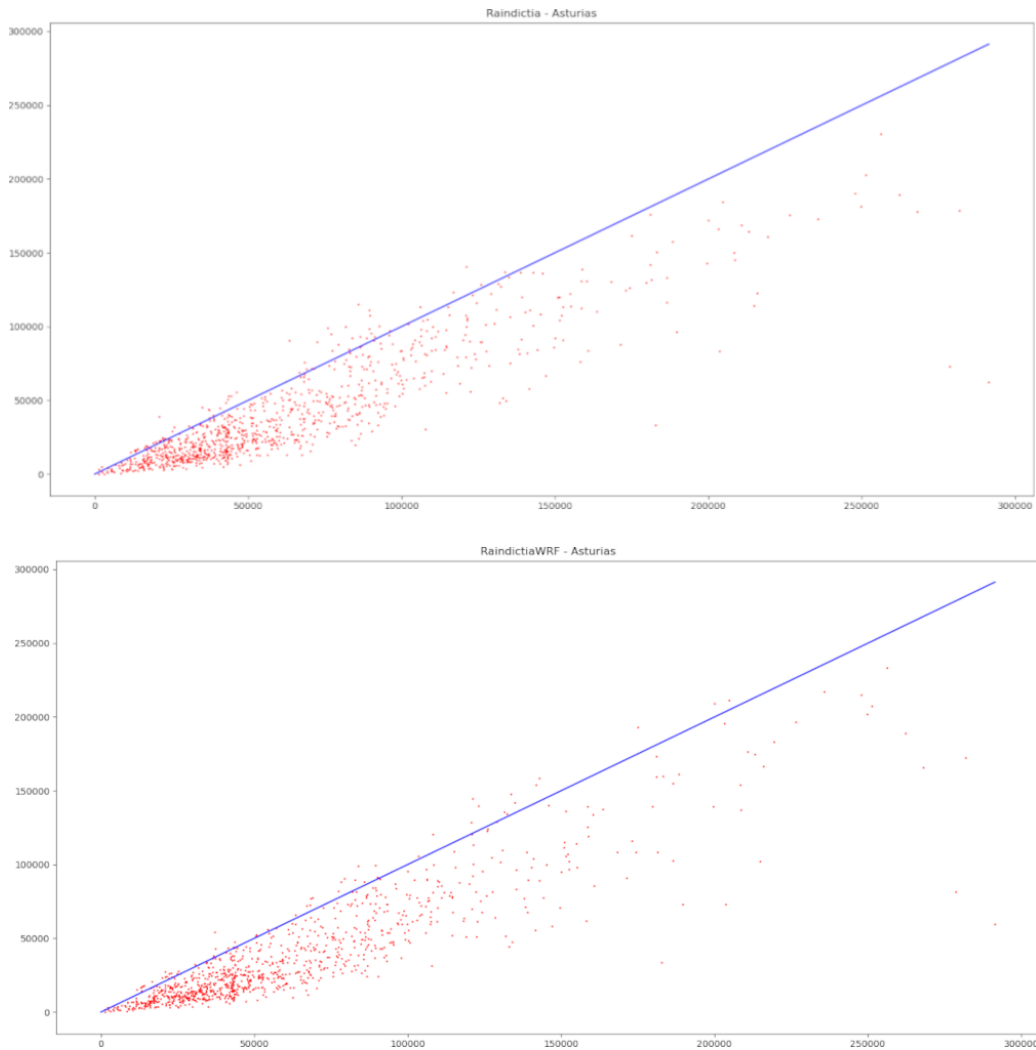


Figura 5.9: Diagrama de dispersión del total de precipitación acumulado espacio-temporalmente predicho con respecto al observado para las muestras del radar de Asturias, mostrando los resultados para los modelos de aprendizaje profundo Raindictia (arriba) y RaindictiaWRF (abajo).



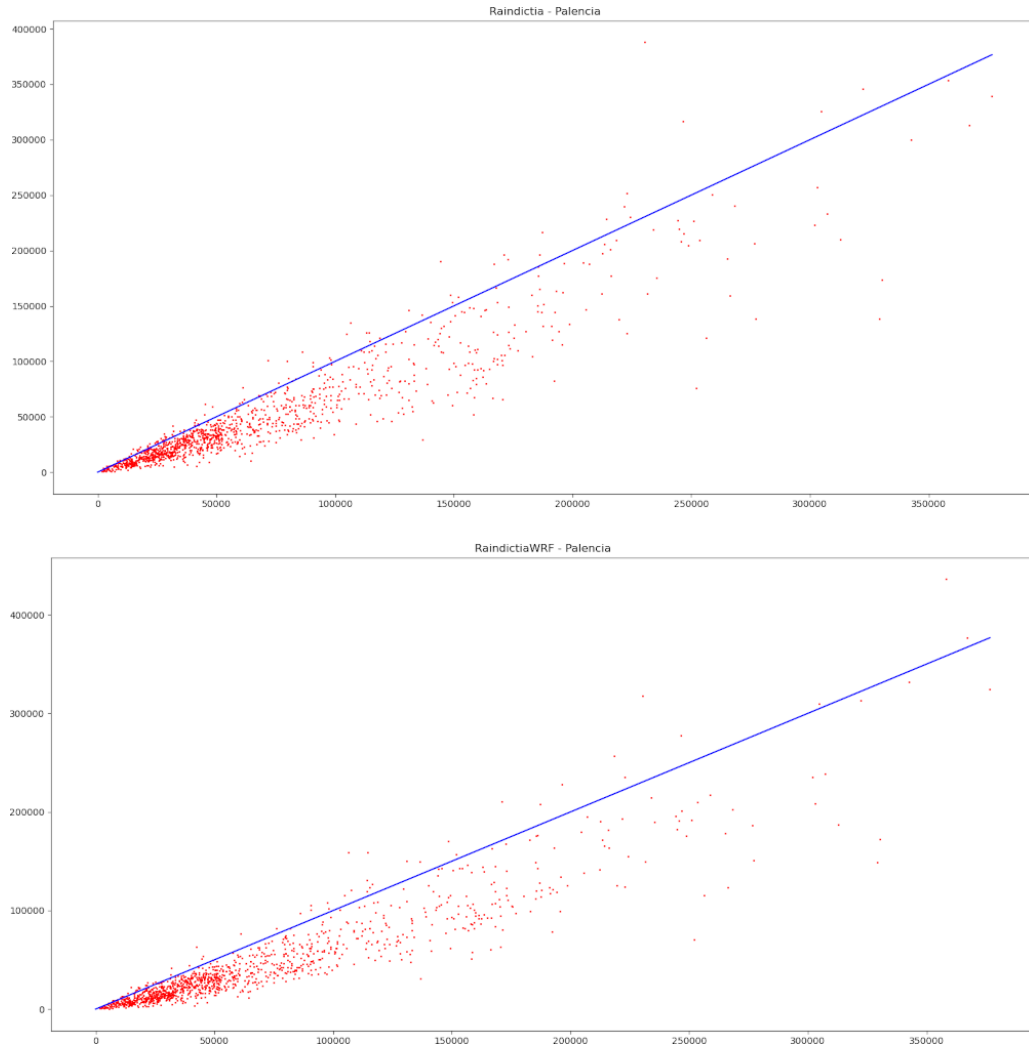


Figura 5.10: Diagrama de dispersión del total de precipitación acumulado espacio-temporalmente predicho con respecto al observado para las muestras del radar de Palencia, mostrando los resultados para los modelos de aprendizaje profundo Raindictia (arriba) y RaindictiaWRF (abajo).

### 5.3. Evaluación sobre casos de estudio

De acuerdo a lo explicado anteriormente no deberían existir muestras con menos de un 5 % de valores distintos de 0, y así se ha cumplido. Sin embargo, se ha comprobado que existen muestras en las que dicho 5 % de valores se corresponden con ruido producido por el radar (ver Figura 5.11). Estas muestras producen unos datos de evaluación que se alejan mucho del promedio (ver Tabla 5.4) y que pueden haber influenciado las tablas globales mostradas anteriormente, tanto reduciendo el error (como puede ser el caso de RMSE y MAE) como aumentándolo (CSI y Correlación).

Tiempo de predicción	RMSE	MAE	CSI	Correlation
t+10min	0.07	0.008	0.33	0.41
t+20min	0.07	0.007	0.19	0.24
t+30min	0.06	0.006	0.17	0.23
t+40min	0.07	0.007	0.13	0.16
t+50min	0.08	0.007	0.07	0.06
t+60min	0.08	0.008	0.04	0.05

Tabla 5.4: Métricas de evaluación para la predicción sobre unos datos de entrada formados principalmente por ruido que proceden de una de las muestras de test.

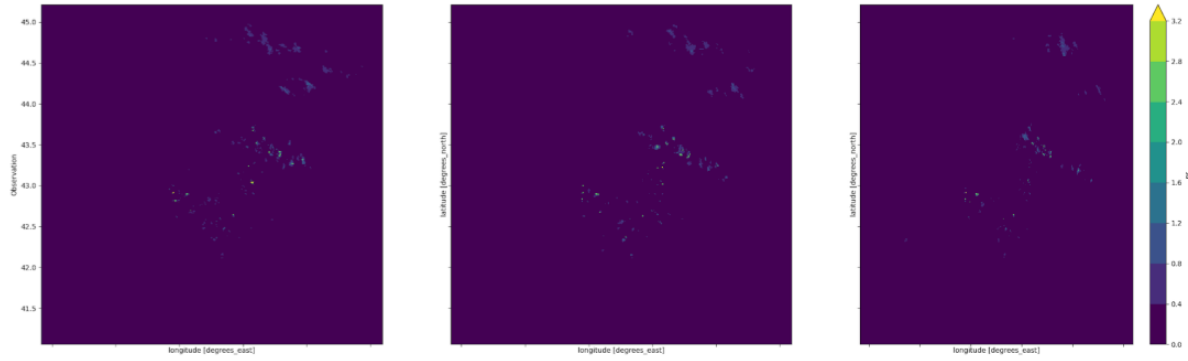


Figura 5.11: Observación formada básicamente por ruido utilizada para comparar con la predicción de los modelos en una muestra de test.

Por ello, se ha decidido dedicar esta sección a la visualización y evaluación de determinadas muestras de estudio donde la mancha de lluvia presenta un patrón característico tanto en magnitud como en localización. Así se podrá ver gráficamente cual es la actuación de nuestros modelos en el estudio espacio-temporal del campo de precipitación en casos donde los datos de entrada del modelo contengan un cantidad de precipitación considerable, y no se vean perjudicados porque están formados principalmente por ruido.

Un ejemplo de este tipo de muestras viene dado por la observación del radar de Palencia correspondiente al 13 de Noviembre de 2019 a las 00:50 UTC (ver Figura 5.12). En este caso se predice, mediante el uso de Raindictia, de manera notable la localización de la mancha de precipitación y su desplazamiento. Además su magnitud (aunque algo suavizada) parece representada con suficiente precisión a lo largo de los diferentes horizontes de predicción. Dentro de la mancha de precipitación podemos distinguir dos patrones diferentes: a la izquierda una mancha más pequeña y la derecha encontramos donde estaría el patrón más grande en espacio y magnitud. El modelo de aprendizaje automático es capaz de mantener ambos patrones en el tiempo, aunque el patrón situado a la izquierda pierde magnitud con mayor rapidez.

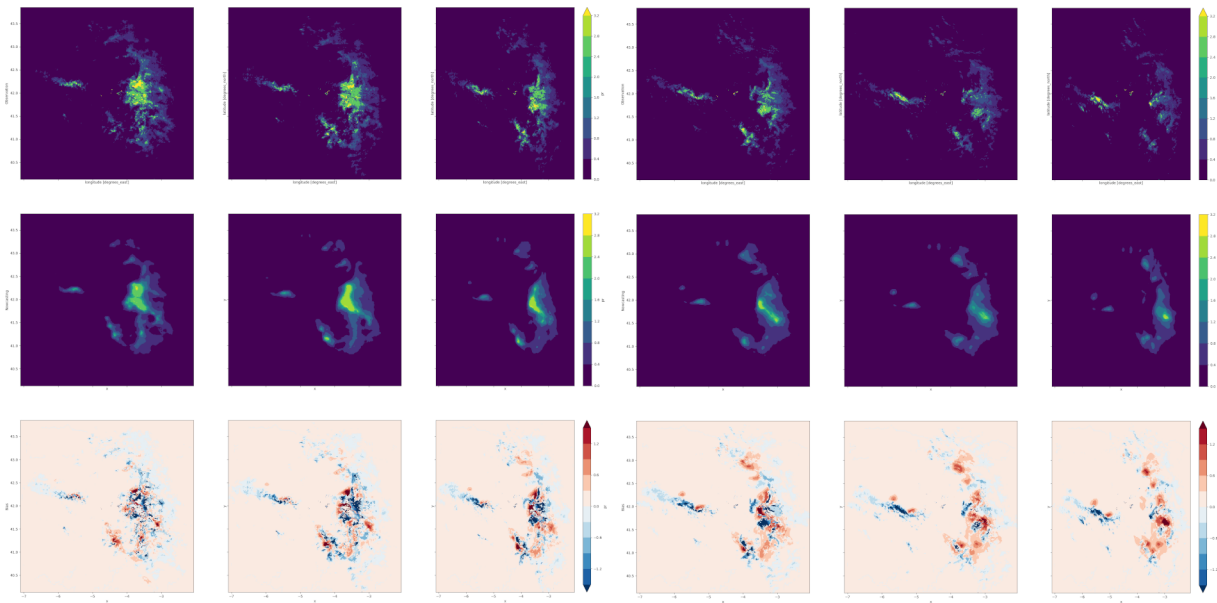


Figura 5.12: Observación, predicción y sesgo (de arriba a abajo) para una muestra de Palencia correspondiente al 13 de Noviembre de 2019 a las 00:50 UTC, estudiando los horizontes de predicción diezminutales desde  $t+10\text{min}$  hasta  $t+60\text{min}$  (de izquierda a derecha). La predicción se ha realizado con el modelo Raindictia.

Esto es posible distinguirlo también desde el punto de vista de las métricas obtenidas como resultado de validación de la muestra. El CSI tiene un rango desde 0.73 hasta 0.45, mientras que la correlación comienza en 0.80 y finaliza en 0.38 (valores desde  $t+10\text{min}$  hasta  $t+60\text{min}$ ). Estos valores superan con abundancia a los promedios obtenidos sobre el conjunto de muestras de validación cuyos resultados aparecen en la Tabla 5.3.

Otro caso de estudio que se muestra como un buen ejemplo de muestra que supera con creces el promedio dado en las tablas del inicio del capítulo es el de la Figura 5.13. El patrón espacial se sigue claramente, incluso se puede apreciar una línea de interferencia del radar, que parte el patrón en dos partes diferentes, que el modelo de aprendizaje profundo es capaz de captar. En este caso la métricas también acompañan ya que el CSI tiene va desde 0.83 hasta 0.61, mientras que la correlación se mueve desde entorno 0.89 hasta 0.55 (valores desde  $t+10\text{min}$  hasta  $t+60\text{min}$ ).

Hay que destacar que no es una muestra en la que la mancha de precipitación tenga un gran desplazamiento a medida que pasa el tiempo, parece que se corresponde con una *lluvia frontal* donde una masa de aire cálido y una masa de aire frío entran en contacto.

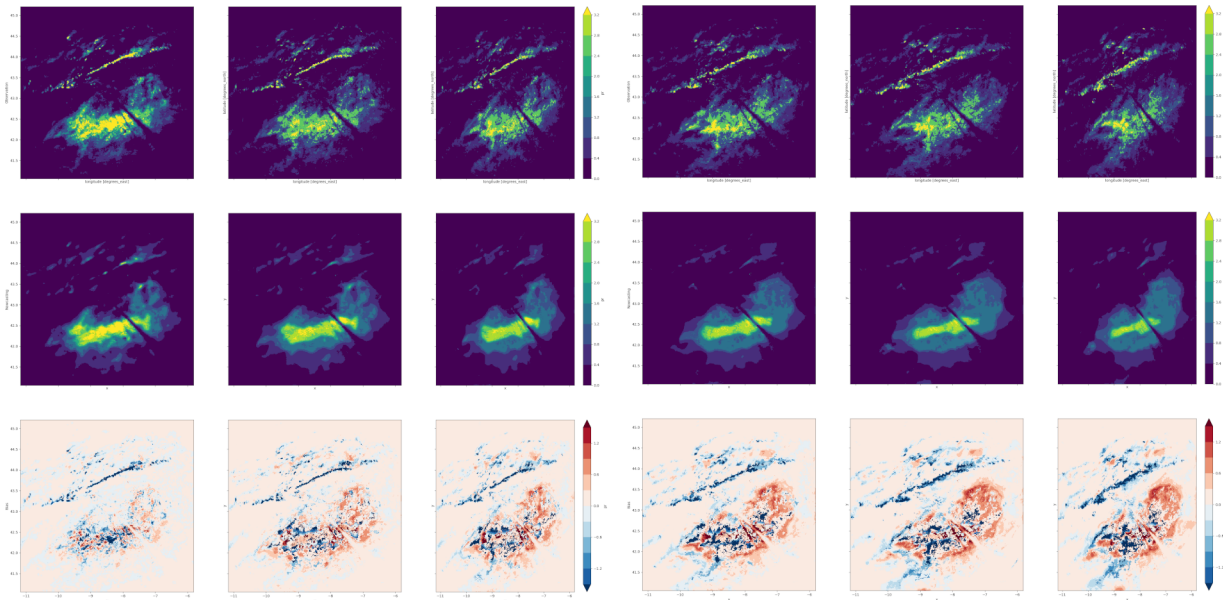


Figura 5.13: Observación, predicción y sesgo (de arriba a abajo) para una muestra de A Coruña correspondiente al 19 de Octubre de 2019 a las 07:50 UTC, estudiando los horizontes de predicción diezminutales desde  $t+10\text{min}$  hasta  $t+60\text{min}$  (de izquierda a derecha). La predicción se ha realizado con el modelo Raindictia.

Estos ejemplos son numerosos, lo que quiere decir que existen muestras que empeoran los resultados globales de validación considerablemente. Tal vez otro tipo de validación global basada en la mediana o algún percentil desvanecería el efecto de las muestras sobre las que se hace una mala predicción.

#### 5.4. Efecto del cuantil elegido en la función de coste

Para comprobar el efecto del cuantil de la función de coste, sobre la predicciones de nuestro modelo, se han entrenado varios modelos modificando este parámetro de la función de pérdida. Ambos modelos han sido entrenados sobre los datos del radar regional de Asturias y con la arquitectura de Raindictia (sin añadir la variable predictora conformada por el modelo numérico WRF). El primer modelo con cuantil igual a 0.55 (ver Figura 5.14) presenta una modelización del patrón espacial robusta, tal vez un poco por debajo del valor que nos gustaría. Mientras tanto, el segundo con cuantil igual a 0.60 (ver Figura 5.15) presenta una desviación enorme a partir de la cuarta iteración.

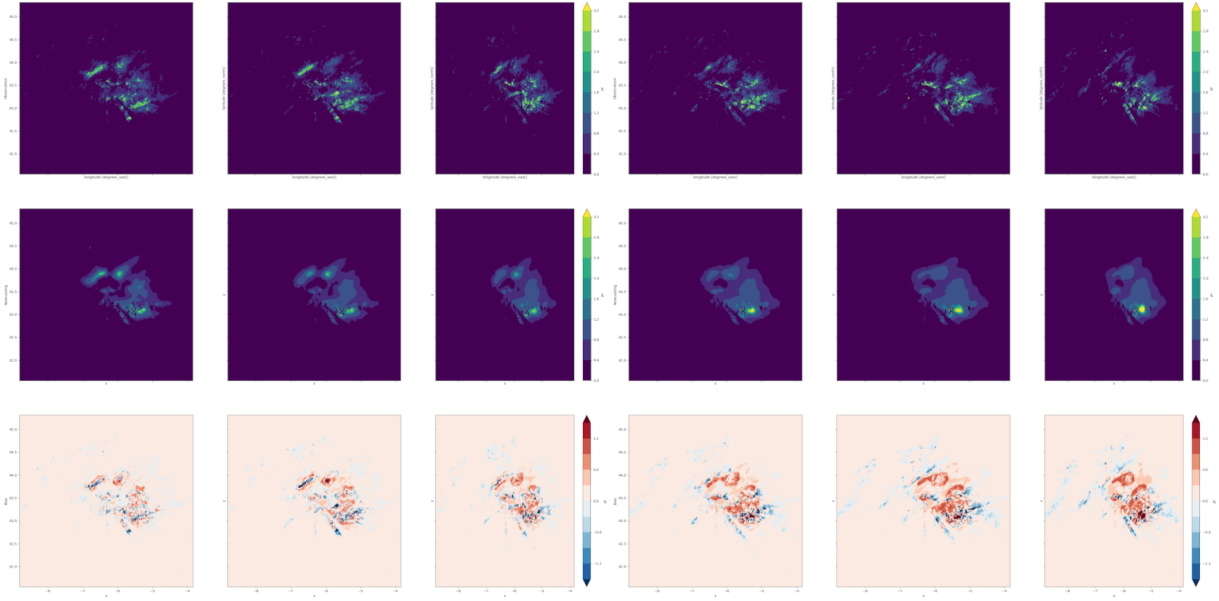


Figura 5.14: Observación, predicción y sesgo (de arriba a abajo) para una muestra de Asturias correspondiente al 19 de Octubre de 2019 a las 14:20 UTC, estudiando los horizontes de predicción diezminutales desde  $t+10\text{min}$  hasta  $t+60\text{min}$  (de izquierda a derecha). El modelo entrenado utilizó una función de coste como la de la Ecuación 4.5 con el parámetro igual a 0.55.

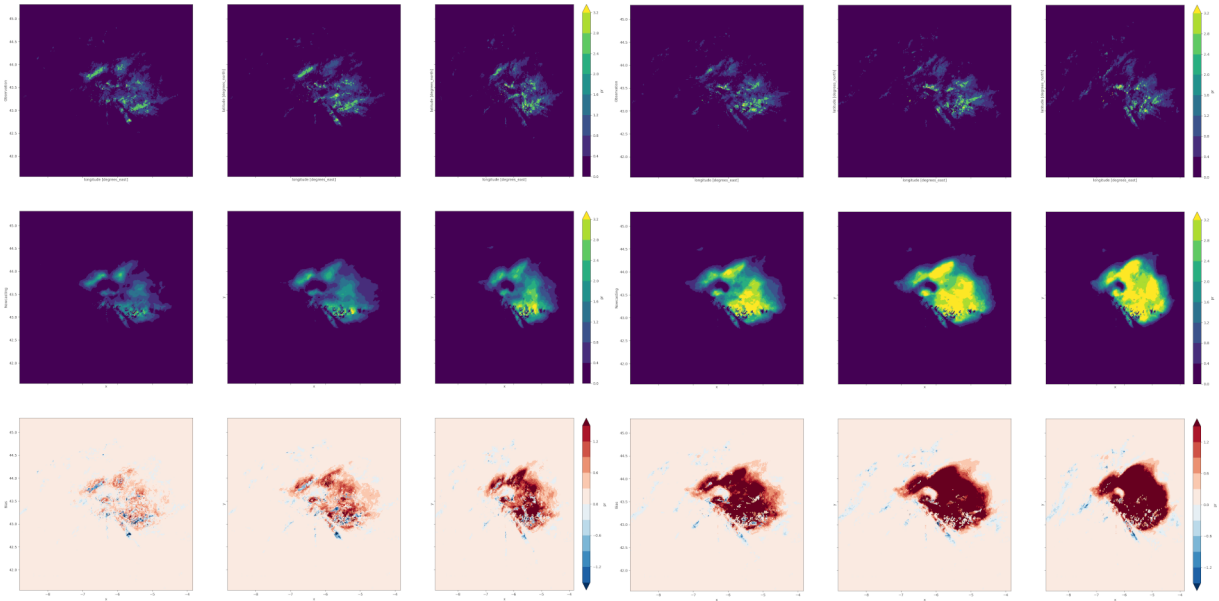


Figura 5.15: Observación, predicción y sesgo (de arriba a abajo) para una muestra de Asturias correspondiente al 19 de Octubre de 2019 a las 14:20 UTC, estudiando los horizontes de predicción diezminutales desde  $t+10\text{min}$  hasta  $t+60\text{min}$  (de izquierda a derecha). El modelo entrenado utilizó una función de coste como la de la Ecuación 4.5 con el parámetro igual a 0.60.

Como se puede imaginar la elección del cuantil no es una decisión sencilla y requiere de mucho tiempo y búsqueda para encontrar el que mejor optimiza el periodo de entrenamiento. Igual por esto, Google desarrollo el problema como un problema de clasificación y no de regresión. En el caso de *MetNet*, red neuronal desarrollada por Google para resolver el problema de nowcasting de precipitación, se desarrolla en forma de problema multicategoría (512 categorías), donde cada categoría representa un rango de 0.2 mm/h de precipitación. Tal vez así, hubiéramos solucionado este problema.

## Parte IV

# Conclusiones

## Capítulo 6

# Conclusiones y trabajos futuros

Cuando se inició el proyecto en Octubre de 2019 no existía apenas información sobre el problema de nowcasting de precipitación, a parte de aquellos artículos que versaban sobre el uso de técnicas basadas en la persistencia lagrangiana y el gran aporte realizado por Xingjian Shi con la creación de las capas Conv-LSTM para tratar de modelizar el problema. Sin embargo, en la revisión bibliográfica que se ha realizado a la finalización del mismo se presentan abundantes referencias a trabajos que plantean el problema desde el punto de vista del aprendizaje profundo. En especial dos instituciones, como Google LLC y MeteoSwiss, han creado modelos de aprendizaje profundo para resolver este problema, lo que se traduce en una forma de valía hacia la solución del problema considerado debido a sus posibles aplicaciones.

Por la parte que nos toca, el objetivo principal del trabajo se ha completado. Durante el tiempo de investigación, se ha desarrollado un software capaz de llevar a cabo la descarga, almacenamiento y procesamiento de datos procedentes tanto de una fuente externa como es la Agencia Estatal de Meteorología (AEMET) como de simulaciones numéricas convencionales realizadas en el entorno empresarial de Predictia Intelligent Data Solutions S.L.

No solo esto, sino que también se han analizado dichos datos y se han generado varios modelos que permiten predecir la variabilidad espacio-temporal de la precipitación. Dichos modelos se basan en una estructura de capas convolucionales que primeramente decrecen la dimensionalidad espacial de los datos de entrada para posteriormente devolverlos a su estructura original. Raindictia y RaindictiaWRF son muy similares en arquitectura, teniendo como principal diferencia el tamaño de los datos de entrada. En el primer caso se utilizan datos procedentes únicamente de las observaciones de radar, mientras que en el segundo se combinan dichas observaciones con datos que provienen de modelos numéricos convencionales (NWP).

Para determinar si unos resultados son "notables" o no, se ha llevado a cabo un proceso de validación donde se han estudiado las estructuras de redes neuronales en tres radares regionales de la red de AEMET: A Coruña, Asturias y Palencia. Todos los modelos han sido entrenados de la misma forma y parecen converger de una manera muy similar. Sin embargo, en el caso de RaindictiaWRF para el radar de A Coruña el modelo es inestable. Este proceso de validación hace uso de un gran número de muestras, concretamente sobrepasa el número de muestras empleado por Rainnet en algo más de un 1000 %.

De acuerdo a las métricas de evaluación, tanto Raindictia como RaindictiaWRF superan a la persistencia euleriana y a algunas técnicas estadísticas (extrapolación y STEPS) en todos los radares. Sin embargo, al compararlos con la técnica de modelización espectral S-PROG no es tan claro designar que modelo es más preciso que otro. En principio, los dos modelos de aprendizaje profundo se encuentran por encima en términos del CSI, MAE y RMSE, pero la correlación de S-PROG parece superar a la de los modelos de aprendizaje profundo en todos los horizontes de predicción.

Además, se ha comprobado de varias formas diferentes que los modelos de aprendizaje profundo llevan a cabo un suavizado de la mancha de precipitación, asociado con el proceso de optimización y la elección de la función de coste, que produce que la precipitación total predicha en términos espacio-temporales se desvíe en gran medida con respecto a la observación real para un determinado horizonte de predicción.

Concluimos que Raindictia y RaindictiaWRF se comportan mejor que todas las técnicas usadas como referencia en términos de RMSE, MAE y CSI, al igual que ocurre en la validación llevada a cabo por Rainnet. Sin embargo, S-PROG presenta un mejor desempeño en términos de correlación (esta métrica no fue incluida en la validación de otros modelos en el ámbito del aprendizaje automático).

Por tanto, se han obtenido valores que casan con lo esperado antes de comenzar los experimentos. Es cierto que se pudiera haber realizado un estudio más a fondo de algunos parámetros de la red neuronal (funciones de activación, algoritmo de optimización y función de coste) y de las transformaciones realizadas a los datos, pero también se debe tener en cuenta la sencillez en la estructura de la red utilizada.

A partir de una estructura de red sencilla (no llega a 500 mil parámetros) y de un conjunto de datos de entrenamiento ni mucho menos elevado (ver Tablas 4.1 y 4.2) se han obtenido resultados prometedores tanto para Raindictia como para RaindictiaWRF. Previamente al diseño de los modelos, no se esperaba que RaindictiaWRF fuera a producir mejores resultados que Raindictia a escalas temporales muy bajas (las primeras horas de predicción), sino que se esperaba que el valor añadido de la simulación numérica de mesoescala (WRF) apareciera para periodos de tiempo más elevados. Esto no se ha podido comprobar al realizar una evaluación sobre el conjunto de datos de test de una predicción a 60 minutos.

Otros modelos predictivos de precipitación a muy corto plazo, basados en el uso de observaciones de radar y otras técnicas de teledetección, que han sido desarrollados en el ámbito empresarial y de investigación (RainNet, MetNet, ...) presentan estructuras de millones de parámetros que no se acercan ni de lejos a los apenas 500 mil parámetros utilizados con Raindictia y RaindictiaWRF. Además, utilizan un mayor número de datos de entrenamiento, a la par que un menor número de muestras de validación.

Por tanto, se deja para el futuro el aumentar tanto las muestras utilizadas para el entrenamiento de los modelos como el tiempo de predicción en la validación de los experimentos. Además, se espera desarrollar un método para la optimización del cuantil elegido en la función de coste o incluso elegir otra función de coste que siga el mismo esquema (funciones a trozos).

Para evitar esto se pudiera plantear el problema de nowcasting con un esquema de clasificación multicategoría en lugar de mantener el enfoque de regresión que se le ha dado durante este trabajo. Utilizar un esquema de clasificación permitiría el olvidarse del problema de la función de coste que se ha comentado a lo largo de toda la memoria.

No todos los problemas provienen de la función de coste, sino que muchos de las desviaciones mostradas en los resultados se deben a la concatenación de errores debido al esquema de predicción iterativa utilizado. Por ello, otro posible factor de estudio sería implementar una técnica de predicción más estable que no arrastre errores a través de los diferentes horizontes de predicción. Directamente, se pudiera aumentar el tamaño de la matriz de salida del modelo y que automáticamente el modelo fuera capaz de predecir más que un único tiempo de predicción.



# Bibliografía

- [1] American Meteorological Society. **Precipitation - Glossary of Meteorology**. URL: <http://glossary.ametsoc.org/wiki/Precipitation> (visitado 18-08-2020).
- [2] Rebecca E. Morss y col. **Improving Societal Outcomes of Extreme Weather in a Changing Climate: An Integrated Perspective**. En: *Annual Review of Environment and Resources* 36.1 (2011), págs. 1-25. DOI: [10.1146/annurev-environ-060809-100145](https://doi.org/10.1146/annurev-environ-060809-100145).
- [3] World Meteorological organization (WMO). **Nowcasting**. URL: <https://www.wmo.int/pages/prog/amp/pwsp/Nowcasting.htm> (visitado 18-08-2020).
- [4] Peter Bauer, Alan Thorpe y Gilbert Brunet. **The quiet revolution of numerical weather prediction**. En: *Nature* 525.7567 (sep. de 2015), págs. 47-55. ISSN: 1476-4687. DOI: [10.1038/nature14956](https://doi.org/10.1038/nature14956).
- [5] Wansik Yu y col. **Improvement of rainfall and flood forecasts by blending ensemble NWP rainfall with radar prediction considering orographic rainfall**. En: *Journal of Hydrology*. Hydrologic Applications of Weather Radar 531 (dic. de 2015), págs. 494-507. ISSN: 0022-1694. DOI: [10.1016/j.jhydrol.2015.04.055](https://doi.org/10.1016/j.jhydrol.2015.04.055).
- [6] Urs Germann, Isztar Zawadzki y Barry Turner. **Predictability of Precipitation from Continental Radar Images. Part IV: Limits to Prediction**. En: *J. Atmos. Sci.* 63.8 (ago. de 2006). Publisher: American Meteorological Society, págs. 2092-2108. ISSN: 0022-4928. DOI: [10.1175/JAS3735.1](https://doi.org/10.1175/JAS3735.1).
- [7] M. Reyniers e Institut royal météorologique de Belgique. **Quantitative Precipitation Forecasts Based on Radar Observations: Principles, Algorithms and Operational Systems**. Publication scientifique et technique. Institut Royal Météorologique de Belgique, 2008. URL: <https://books.google.es/books?id=NgNsPgAACAAJ>.
- [8] Isztar Zawadzki. **Factors Affecting the Precision of Radar Measurements of Rain**. En: *Conference on Radar Meteorology of the American Meteorological Society* (ene. de 1984), págs. 251-256.
- [9] Agencia Estatal de Meteorología AEMET. Gobierno de España. **Observaciones de teledetección terrestre**. URL: <http://www.aemet.es/es/idi/observacion/teledeteccion> (visitado 20-08-2020).
- [10] M. Heistermann, S. Jacobi y T. Pfaff. **Technical Note: An open source library for processing weather radar data wradlib**. En: *Hydrol. Earth Syst. Sci.* 17.2 (feb. de 2013), págs. 863-871. ISSN: 1607-7938. DOI: [10.5194/hess-17-863-2013](https://doi.org/10.5194/hess-17-863-2013).
- [11] Georgy Ayzel, Maik Heistermann y Tanja Winterrath. **Optical flow models as an open benchmark for radar-based precipitation nowcasting (rainymotion v0.1)**. En: *Geoscientific Model Development* 12.4 (abr. de 2019). Publisher: Copernicus GmbH, págs. 1387-1402. ISSN: 1991-959X. DOI: <https://doi.org/10.5194/gmd-12-1387-2019>.



- [12] Seppo Pulkkinen y col. **Pysteps: an open-source Python library for probabilistic precipitation nowcasting (v1.0)**. En: *Geoscientific Model Development* 12.10 (oct. de 2019). Publisher: Copernicus GmbH, págs. 4185-4219. ISSN: 1991-959X. DOI: <https://doi.org/10.5194/gmd-12-4185-2019>.
- [13] Alejandro Buil Martínez. **Nowcasting probabilístico basado en observaciones de lluvia con radar meteorológico**. Accepted: 2017-11-22 Publication Title: TDX (Tesis Doctorals en Xarxa). Ph.D. Thesis. Universitat Politècnica de Catalunya, jul. de 2017. URL: <http://www.tdx.cat/handle/10803/456895>.
- [14] Ian Goodfellow, Yoshua Bengio y Aaron Courville. **Deep Learning**. <http://www.deeplearningbook.org>. MIT Press, 2016.
- [15] Sepp Hochreiter y Jürgen Schmidhuber. **Long Short-Term Memory**. En: *Neural Computation* 9.8 (nov. de 1997). Publisher: MIT Press, págs. 1735-1780. ISSN: 0899-7667. DOI: [10.1162/neco.1997.9.8.1735](https://doi.org/10.1162/neco.1997.9.8.1735).
- [16] Xingjian Shi y col. **Convolutional LSTM Network: a machine learning approach for precipitation nowcasting**. En: (sep. de 2015). URL: <https://arxiv.org/abs/1506.04214>.
- [17] Xingjian Shi y col. **Deep Learning for Precipitation Nowcasting: A Benchmark and A New Model**. En: *arXiv[cs]* (oct. de 2017). URL: <http://arxiv.org/abs/1706.03458>.
- [18] Quang-Khai Tran y Sa-kwang Song. **Computer Vision in Precipitation Nowcasting: Applying Image Quality Assessment Metrics for Training Deep Neural Networks**. En: *Atmosphere* 10.5 (mayo de 2019). Number: 5 Publisher: Multidisciplinary Digital Publishing Institute, pág. 244. DOI: [10.3390/atmos10050244](https://doi.org/10.3390/atmos10050244).
- [19] G. Ayzel y col. **All convolutional neural networks for radar-based precipitation nowcasting**. En: *Procedia Computer Science* 150 (ene. de 2019), págs. 186-192. ISSN: 1877-0509. DOI: [10.1016/j.procs.2019.02.036](https://doi.org/10.1016/j.procs.2019.02.036).
- [20] Georgy Ayzel, Tobias Scheffer y Maik Heistermann. **RainNet v1.0: a convolutional neural network for radar-based precipitation nowcasting**. En: *Geoscientific Model Development* 13.6 (jun. de 2020). Publisher: Copernicus GmbH, págs. 2631-2644. ISSN: 1991-959X. DOI: <https://doi.org/10.5194/gmd-13-2631-2020>.
- [21] Olaf Ronneberger, Philipp Fischer y Thomas Brox. **U-Net: Convolutional Networks for Biomedical Image Segmentation**. En: *arXiv[cs]* (mayo de 2015). URL: <http://arxiv.org/abs/1505.04597>.
- [22] Vijay Badrinarayanan, Alex Kendall y Roberto Cipolla. **SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation**. En: *arXiv[cs]* (oct. de 2016). URL: <http://arxiv.org/abs/1511.00561>.
- [23] Shreya Agrawal y col. **Machine Learning for Precipitation Nowcasting from Radar Images**. En: *arXiv[cs, stat]* (dic. de 2019). URL: <http://arxiv.org/abs/1912.12132>.
- [24] Casper Kaae Sønderby y col. **MetNet: A Neural Weather Model for Precipitation Forecasting**. En: *arXiv[physics, stat]* (mar. de 2020). URL: <http://arxiv.org/abs/2003.12140>.
- [25] American Meteorological Society. **Mesoscale model - Glossary of Meteorology**. URL: [http://glossary.ametsoc.org/wiki/Mesoscale\\_model](http://glossary.ametsoc.org/wiki/Mesoscale_model) (visitado 25-08-2020).
- [26] V. Fernández-Quiruelas y col. **WRF4G: WRF experiment management made simple**. En: *Geosci. Model Dev. Discuss.* 8.8 (ago. de 2015), págs. 6551-6582. ISSN: 1991-962X. DOI: [10.5194/gmdd-8-6551-2015](https://doi.org/10.5194/gmdd-8-6551-2015).

- [27] Ju-Hye Kim, Dong-Bin Shin y Christian Kummerow. **Impacts of A Priori Databases Using Six WRF Microphysics Schemes on Passive Microwave Rainfall Retrievals**. En: *J. Atmos. Oceanic Technol.* 30.10 (oct. de 2013). Publisher: American Meteorological Society, págs. 2367-2381. ISSN: 0739-0572. DOI: [10.1175/JTECH-D-12-00261.1](https://doi.org/10.1175/JTECH-D-12-00261.1).
- [28] Eric W. Weisstein. **Lambert Conformal Conic Projection**. Publisher: Wolfram Research, Inc. URL: <https://mathworld.wolfram.com/LambertConformalConicProjection.html> (visitado 25-08-2020).
- [29] Andreas Hackeloeer y col. **Georeferencing: a review of methods and applications**. En: *Annals of GIS* 20.1 (2014), págs. 61-69. DOI: [10.1080/19475683.2013.868826](https://doi.org/10.1080/19475683.2013.868826).
- [30] National Center for Atmospheric Research Staff (Eds). **The Climate Data Guide: Regridding Overview**. Last modified 13 Jan 2014. URL: <https://climatedataguide.ucar.edu/climate-data-tools-and-analysis/regridding-overview> (visitado 02-09-2020).
- [31] Marc Berenguer y col. **Hydrological Validation of a Radar-Based Nowcasting Technique**. En: *J. Hydrometeor.* 6.4 (ago. de 2005). Publisher: American Meteorological Society, págs. 532-549. ISSN: 1525-755X. DOI: [10.1175/JHM433.1](https://doi.org/10.1175/JHM433.1).
- [32] Simon Baker y Iain Matthews. **Lucas-Kanade 20 Years On: A Unifying Framework**. En: *International Journal of Computer Vision* 56.3 (feb. de 2004), págs. 221-255. ISSN: 0920-5691. DOI: [10.1023/B:VISI.0000011205.11775.fd](https://doi.org/10.1023/B:VISI.0000011205.11775.fd).
- [33] A. W. Seed. **A Dynamic and Spatial Scaling Approach to Advection Forecasting**. En: *J. Appl. Meteor.* 42.3 (mar. de 2003). Publisher: American Meteorological Society, págs. 381-388. ISSN: 0894-8763. DOI: [10.1175/1520-0450\(2003\)042<0381:ADASSA>2.0.CO;2](https://doi.org/10.1175/1520-0450(2003)042<0381:ADASSA>2.0.CO;2).
- [34] The Pennsylvania State University Eberly College of Science. **Autoregressive Models**. URL: <https://online.stat.psu.edu/stat501/book/export/html/996> (visitado 04-09-2020).
- [35] Neill Bowler, Clive Pierce y Alan Seed. **STEPS: A probabilistic precipitation forecasting scheme which merges an extrapolation nowcast with downscaled NWP**. En: *Quarterly Journal of the Royal Meteorological Society* 132 (ene. de 2007), págs. 2127 -2155. DOI: [10.1256/qj.04.100](https://doi.org/10.1256/qj.04.100).
- [36] Kyunghyun Cho y col. **On the Properties of Neural Machine Translation: Encoder-Decoder Approaches**. En: *arXiv[cs, stat]* (oct. de 2014). URL: <http://arxiv.org/abs/1409.1259>.
- [37] Rupesh K Srivastava, Klaus Greff y Jürgen Schmidhuber. **Training Very Deep Networks**. En: *Advances in Neural Information Processing Systems 28*. Ed. por C. Cortes y col. Curran Associates, Inc., 2015, págs. 2377-2385. URL: <http://papers.nips.cc/paper/5850-training-very-deep-networks.pdf>.
- [38] Nitish Srivastava y col. **Dropout: A Simple Way to Prevent Neural Networks from Overfitting**. En: *Journal of Machine Learning Research* 15.56 (2014), págs. 1929-1958. URL: <http://jmlr.org/papers/v15/srivastava14a.html>.
- [39] Pengfei Chen, Guangyong Chen y Shengyu Zhang. **Log Hyperbolic Cosine Loss Improves Variational Auto-Encoder**. En: (sep. de 2018). URL: <https://openreview.net/forum?id=rkglvsC9Ym>.
- [40] Diederik P. Kingma y Jimmy Ba. **Adam: A Method for Stochastic Optimization**. En: *arXiv[cs]* (ene. de 2017). URL: <http://arxiv.org/abs/1412.6980>.
- [41] I.T. Jolliffe y D.B. Stephenson. **Forecast Verification: A Practitioner's Guide in Atmospheric Science**. Wiley, 2003. ISBN: 9780470864418. URL: <https://books.google.es/books?id=cjS9kK8IWbwC>.